

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

## **Diplomová práce**

2013

Jakub Ribka

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

**Porovnání rámců pro vývoj webových  
aplikací na platformě Java EE**  
**Comparison of Frameworks for Web  
Application Development in Java EE**

## Zadání diplomové práce

Student:

**Bc. Jakub Ribka**

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Porovnání rámců pro vývoj webových aplikací na platformě Java EE  
Comparison of Frameworks for Web Application Development in Java EE

Zásady pro vypracování:

Cílem práce je zhodnotit možnosti vybraných rámců pro vývoj webových aplikací na platformě Java EE. Práce se zaměří na vytvoření aplikace, která bude obsahovat všechny běžné vlastnosti webových aplikací jako je přihlášení, přístup k databázi, a další. Aplikace bude vytvořena ve všech zvolených rámcích, přičemž bude vyhodnocena:

1. Složitost a rychlost vývoje jednotlivých prvků aplikace.
2. Náročnost prvního seznámení s technologií.
3. Dostupnost a přehlednost dokumentace.
4. Podpora v nástrojích pro vývoj.
5. Výkonnost vzniklé aplikace na základě zátěžových a výkonových testů.
6. Podpora technologie AJAX a uživatelská "přítulnost".

V práci porovnejte rámce Java Server Faces, Vaadin, Struts, Spring.

Seznam doporučené odborné literatury:

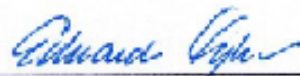
- [1] JENDROCK, Eric, Ian EVANS, Devika GOLLAPUDI, Kim HAASE, William Markito OLIVEIRA a Chinmayee SRIVATHSA. ORACLE. The Java EE 6 Tutorial. 2012. Dostupné z <http://docs.oracle.com/javaee/6/tutorial/doc/javaee6tutorial6.pdf>
- [2] FORD, Neal. Art of Java web development: Struts, Tapestry, Commons, Velocity, JUnit, Axis, Cocoon, InternetBeans, WebWorks. Greenwich, CT: Manning, c2004, xxx, 593 p. ISBN 19-323-9406-0.
- [3] GRÖNROOS, Marko. Book of Vaadin. 4th ed. Turku, Finland: Vaadin Ltd, 2012. ISBN 978-952-9267-538. Dostupné z: <https://vaadin.com/book/-/page/index.html>

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. David Ježek, Ph.D.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry





prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

### **Poděkování**

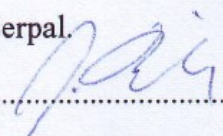
Na tomto místě bych rád poděkoval vedoucímu mé diplomové práce Ing. Davidu Ježkovi Ph.D. za velmi cennou pedagogickou i odbornou pomoc při řešení a vstřícný přístup k vedení této práce.

### **Prohlášení**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 15. dubna 2013

.....  
  
Podpis

„Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.“

V Ostravě dne 15. dubna 2013

.....

Podpis

# Abstrakt

Tato diplomová práce se zabývá srovnáním webových frameworků pro programovací jazyk Java. Budu se zde věnovat těmto čtyřem frameworkům: Java Server Faces, Vaadi, Struts 2, Spring MVC. Nejprve se podíváme obecně na webové aplikace v Javě a představíme si jednotlivé frameworky, poté se seznámíme s testovací aplikací, na které bude srovnávání probíhat. V našem případě se bude jednat o malý e-shop. Na závěr proběhne výsledné srovnání a vyhodnocení daných frameworků dle různých kritérií, největší důraz bude kladen na výkonnostní a zátěžové testy.

## Klíčová slova

Java, MySQL, Hibernate, Vaadin, Spring MVC, JSF, Struts2, Framework

## Abstract

This diploma thesis deals with a comparison of web frameworks for programming language Java. I will pursue these four frameworks: Java Server Faces, Vaadin, Struts 2, Spring MVC. First we look at general web applications in Java and introduce the individual frameworks. Then we make familiar the test application which will be compared. In our case it will be a small e-shop. In the final part of the thesis the author brings the resulting comparison and the evaluation of the frameworks according to various criteria. It emphasizes on performance and load tests.

## Key words

Java, MySQL, Hibernate, Vaadin, Spring MVC, JSF, Struts2, Framework

# Seznam použitých symbolů a zkratek

SQL – Sturctured query langure

MySQL – Databázový systém využívající jazyka SQL

HTML - HyperText Markup Language

JSF – Java Server Faces

JSP – Java Server Pages

XHTML – Extensible hypertext markup language

MVC – Model View Controller

# Obsah

1	Úvod .....	1
2	Specifikace cíle.....	2
2.1	Softwarové prostředky .....	2
2.2	Seznam porovnávaných frameworků .....	2
3	Webové aplikace a JAVA – J2EE .....	3
3.1	Historie .....	3
3.2	Model-View-Contreoler .....	3
4	Frameworky v Javě .....	4
4.1	Framework .....	4
4.2	Cíle aplikačních Frameworků .....	4
5	Spring MVC .....	5
5.1	Úvod .....	5
5.2	Vlastnosti.....	5
5.3	Architektura.....	5
5.4	Dokumentace.....	5
5.5	Výhody .....	6
5.6	Nevýhody .....	6
5.7	Shrnutí .....	6
6	Vaadin .....	7
6.1	Úvod .....	7
6.2	Vlastnosti.....	7
6.3	Architektura.....	7
6.4	Dokumentace.....	8
6.5	Výhody .....	8
6.6	Nevýhody .....	8
6.7	Shrnutí .....	8
7	Struts2.....	9
7.1	Úvod .....	9



7.2	Vlastnosti.....	9
7.3	Architektura.....	9
7.4	Dokumentace.....	10
7.5	Výhody .....	10
7.6	Nevýhody .....	10
7.7	Shrnutí .....	11
8	Java Server Faces .....	12
8.1	Úvod.....	12
8.2	Vlastnosti.....	12
8.3	Architektura.....	12
8.4	Dokumentace.....	13
8.5	Výhody .....	13
8.6	Nevýhody .....	14
8.7	Shrnutí .....	14
9	Testovací aplikace .....	15
9.1	Úvod.....	15
9.2	Popis testovací aplikace.....	15
9.2.1	Schéma databáze .....	16
9.2.2	Tabulky.....	16
10	Grafické ukázky v jednotlivých frameworkích .....	19
10.1	Vaadin .....	19
10.1.1	Úvodní strana .....	19
10.1.2	Košík .....	20
10.1.3	Registrace .....	20
10.2	Java Server Faces .....	21
10.2.1	Úvodní strana .....	21
10.2.2	Registrace .....	22
10.2.3	Košík .....	22
10.3	Struts2.....	23

10.3.1	Úvodní strana .....	23
10.3.2	Registrace .....	24
10.3.3	Košík .....	24
10.4	Spring MVC .....	25
10.4.1	Úvodní strana .....	25
10.4.2	Nákupní košík.....	26
10.4.3	Registrace .....	26
11	Srovnání vybraných frameworků .....	27
11.1	Instalace pluginů.....	27
11.1.1	Velikosti jednotlivých pluginů .....	27
11.2	Složitost a rychlost vývoje jednotlivých prvků aplikace .....	27
11.2.1	Vaadin .....	27
11.2.2	Java Server Faces .....	28
11.2.3	Spring MVC .....	28
11.2.4	Struts2.....	28
11.2.5	Vyhodnocení .....	28
11.3	Náročnost prvotního seznámení s technologií.....	29
11.3.1	Vaadin .....	29
11.3.2	Java Server Faces .....	29
11.3.3	Spring MVC .....	29
11.3.4	Struts2.....	29
11.3.5	Vyhodnocení .....	30
11.4	Dostupnost a přehlednost dokumentace .....	30
11.4.1	Vaadin .....	30
11.4.2	Java Server Faces .....	30
11.4.3	Spring MVC .....	31
11.4.4	Struts.....	31
11.4.5	Vyhodnocení .....	31
11.5	Podpora v nástrojích pro vývoj.....	31

11.5.1	Vaadin .....	31
11.5.2	Java Server Faces, Struts2, Spring MVC .....	32
11.5.3	Vyhodnocení .....	32
11.6	Výkonnost vzniklé aplikace na základě zátěžových a výkonnostních testů.....	32
11.6.1	Nástroje a nastavení.....	32
11.6.2	Vaadin .....	33
11.6.3	Java Server Faces .....	35
11.6.4	Spring MVC .....	36
11.6.5	Struts2.....	38
11.6.6	Vyhodnocení .....	40
11.7	Podpora technologie AJAX a uživatelská "přítulnost" .....	40
11.7.1	Vaadin .....	41
11.7.2	Java Server Faces .....	41
11.7.3	Spring MVC .....	41
11.7.4	Struts2.....	41
11.7.5	Vyhodnocení .....	41
11.8	Počet řádků zdrojového kódu a podpůrných souborů .....	41
11.8.1	Vaadin .....	42
11.8.2	Java Server Faces .....	43
11.8.3	Spring MVC .....	43
11.8.4	Struts 2.....	44
11.8.5	Vyhodnocení .....	44
11.9	Srovnání kódu – přihlášení do aplikace.....	44
11.9.1	Vaadin .....	44
11.9.2	Java Server Faces .....	45
11.9.3	Spring MVC .....	47
11.9.4	Struts2.....	48
11.9.5	Vyhodnocení .....	49
12	Celkové vyhodnocení porovnávaných frameworků .....	50

12.1	Slovní hodnocení.....	50
12.1.1	Vaadin .....	50
12.1.2	Java Server Facer.....	50
12.1.3	Spring MVC .....	50
12.1.4	Struts2.....	51
12.2	Tabulkové vyhodnocení .....	51
13	Závěr.....	52
14	Seznam tabulek.....	53
15	Seznam obrázků .....	54

# 1 Úvod

Tématem této diplomové práce je porovnání frameworků (česky: rámců) pro vývoj webových aplikací na platformě Java EE. Konkrétně se bude týkat těchto webových frameworků: Vaadin, Struts2, JSF (Java Server Faces) a Spring MVC. Během svého studia jsem se programovacímu jazyku Java a vývoji webových aplikací pomocí tohoto jazyku aktivně věnoval, rovněž jsem absolvoval bakalářskou praxi ve firmě Tieto s.r.o. na pozici Java Developer. Ve firmě Tieto jsem pracoval s webovým frameworkem Vaadin, který je v mé diplomové práci rovněž použit.

Tato diplomová práce by měla sloužit k rychlejší orientaci mezi výše zmíněnými frameworky, čtenáři (webovému developerovi) by mělo být usnadněno rozhodování, který konkrétní webový framework použít na tu či onu aplikaci. V první části této diplomové práce se podíváme teoreticky na jednotlivé frameworky a řekneme si něco o samotných webových aplikacích v Javě. V druhé části bude probírána implementace zvoleného příkladu ve všech čtyřech frameworkcích, jako testovací aplikace nám bude sloužit jednoduchý e-shop. A v poslední třetí části se zaměříme na konečné porovnání testovaných frameworků, a to z hlediska podporovaných technologií, dokumentace, rychlosti vývoje, celkové rychlosti a podobně.

## 2 Specifikace cíle

Po přečtení této diplomové práce by měl být webový vývojář schopen posoudit vhodnost frameworku pro použití v dané aplikaci. V ukázkové webové aplikaci (e-shopu), bude použito co nejvíce standardních a rozšířených komponent, se kterými se lze při vývoji setkat. Bude provedeno hodnocení každého frameworku a vyčteny jeho klady a zápory podle různých kritérií a budou provedeny výkonnostní a zátěžové testy.

### 2.1 Softwarové prostředky

Při vývoji této diplomové práce byly použity tyto softwary:

- programovací jazyk Java 7
- vývojové prostředí Netbeans 7.2.1
- webový server Apache Tomcat 6
- webový framework Spring MVC 3.1.1
- webový framework JSF (Java Server Faces) 2.1
- webový framework Struts2
- webový framework Vaadin 6.2.3
- databázový framework Hibernate 3.2.5
- verzovací systém SVN
- testovací prostředí IBM performance Tester

Jako platforma byl zvolen Windows 8 profesional.

### 2.2 Seznam porovnávaných frameworků

- **webový framework Spring MVC [1]**
- **webový framework JSF (Java Server Faces) [2]**
- **webový framework Struts2 [3]**
- **webový framework Vaadin [4]**

### 3 Webové aplikace a JAVA – J2EE

J2EE (neboli Java EE, dříve označovaná jako *Java 2 Enterprise Edition* nebo *J2EE*) [5] je nadstavba nad standardní J2SE, J2EE podporuje vícevrstvou architekturu, je rozdělena na klientskou vrstvu, střední vrstvu a datovou vrstvu. Jednotlivé součásti platformy Java EE jsou definovány pomocí dílčích specifikací.

#### 3.1 Historie

Původní specifikace J2EE byla vytvořena firmou Sun Microsystems zhruba v letech 1999 - 2000. Od J2EE 1.3 je vývoj veden v rámci JCP.

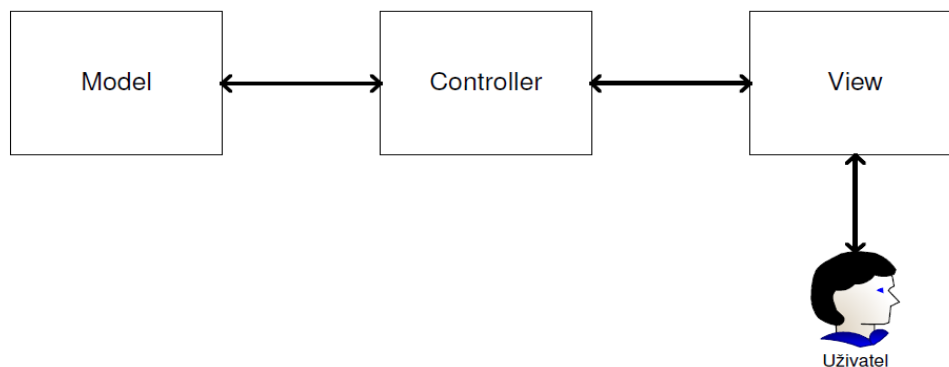
- JSR 58 definuje J2EE 1.3 - finalizována v roce 2001.
- JSR 151 definuje J2EE 1.4 - finalizována v roce 2003.
- JSR 244 definuje Java EE 5 - finalizována v roce 2006.
- JSR 316 definuje Java EE 6 - finalizována v roce 2009.

#### 3.2 Model-View-Contreoler

Jedná se o architekturu [6], která rozděluje datový model aplikace od řídicí logiky a uživatelského rozhraní.

Obecně řečeno vytváření aplikací s využitím architektury MVC vyžaduje vytvoření tří komponent, mezi které patří:

- **Model (model)**, což je doménově specifická reprezentace informací, s nimiž aplikace pracuje.
- **View (pohled)**, který převádí data reprezentovaná modelem do podoby vhodné k interaktivní prezentaci uživateli.
- **Controller (řadič)**, který reaguje na události (typicky pocházející od uživatele) a zajišťuje změny v modelu nebo v pohledu.



Obrázek 1, MVC

## 4 Frameworky v Javě

V mé práci se velmi často vyskytuje pojem framework, proto si tento termín trochu rozebereme, aby bylo jasné, o co se jedná a jakou roli tento pojem v Javě zastává.

### 4.1 Framework

Framework můžeme česky přeložit jako: kostra, skelet, nosná konstrukce, pracovní (aplikační rámec) a podobně. Framework jako takový nelze nějak standardně definovat. Existuje spousta různých názorů a definic pro framework, které se liší různou interpretací.

Nejjednodušeji lze framework charakterizovat jako množinu tříd a rozhraní, které řeší daný problém v určitém programovacím jazyku. V českém jazyce se rovněž používá pojem **rámec**. Framework může rovněž obsahovat různé podpůrné nástroje, API knihovny a implementovat návrhové vzory.

### 4.2 Cíle aplikačních Frameworků

- Obsahovat balíky tříd a rozhraní, kde každá z nich představuje abstrakci nějakého problému
- Definovat jakým způsobem budou tyto abstrakce spolupracovat na vyřešení problému
- Musí umožnit několikanásobnou použitelnost jednotlivých částí (komponent) frameworku
- Umožnit rozšiřitelnost připojováním externích funkcí
- Poskytovat schopnost využití návrhových vzorů



## 5 Spring MVC

### 5.1 Úvod

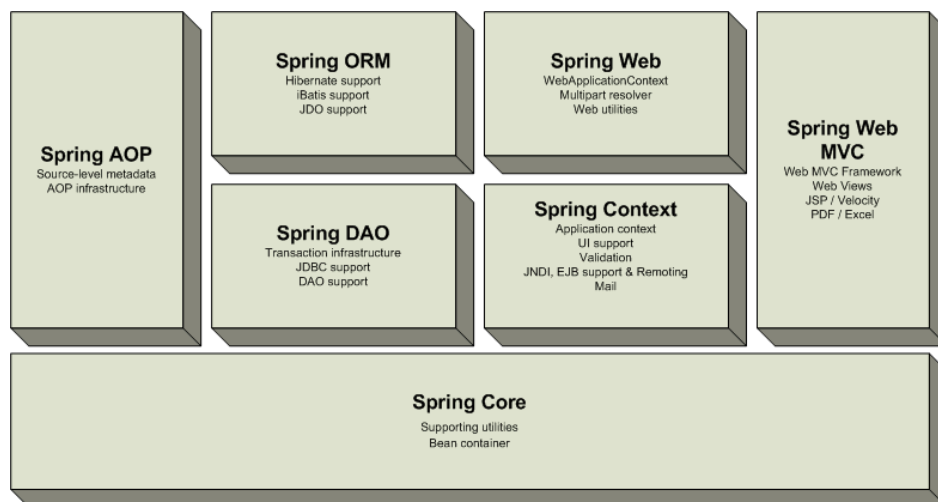
První verze tohoto populárního frameworku byla sepsána v knize Expert One-on-One J2EE Design and Development [7] v říjnu roku 2002 a jejím autorem je Rod Johnson. Framework byl poprvé uvolněn v červnu pod licencí Apache 2.0. První verze 1.0 byla vydaná v březnu 2004, další poté v září 2004 a březnu 2005.

### 5.2 Vlastnosti

Jak již je z názvu patrné jedná se o framework, který podporuje vícevrstvou architekturu MVC. Webová část tohoto frameworku je odvozena z frameworku Struts2 a v mnoha ohledech je s ním podobná, je zde umožněna také migrace ze Struts2. Spring MVC rovněž podporuje ORM frameworky (Hibernate, iBatis, apd.).

### 5.3 Architektura

Ve verzi Spring 1.0 je celkem 7 modulů (Core, DAO, ORM, Context, AOP, Web, Web MVC), s příchodem verze Spring 2.0 již dochází ke spojení modulů Web a Web MVC.



Obrázek 2 - Spring struktúra

Jádro celého frameworku je tvořeno kontejnerem Spring Core [1] .

### 5.4 Dokumentace

Spring MVC poskytuje velmi kvalitní a obsáhlou dokumentaci [8]. Mezi její největší přednosti určitě patří její přehlednost a rozsáhlost. Dokumentace Springu MVC je rozdělena podle verzí frameworku a je možné ji vyhledat ve formátu HTML a PDF. Dokumentace se mi zdála tak obsáhlá a

kvalitně zpracována, že podle mého názoru předčí i mnohé knihy a odborné publikace, které se frameworkem Spring MVC zabývají.

Jak již bylo řečeno, existuje celá řada knih zabývajících se Springem, ale málo knih, které by přímo řešily problematiku Spring MVC a navíc drtivá většina těchto knih nikdy nevyšla v češtině.

## 5.5 Výhody

Použití frameworku Spring MVC přináší řadu výhod. Zde jsou některé z nich:

- jednoduchost a příměrnost použití,
- nezávislost výsledného kódu na aplikačním prostředí,
- nulová či minimální závislost aplikačního kódu na Springu,
- vše v jednom – Spring pokrývá (například na rozdíl od Struts) všechny vrstvy aplikace – od prezentační až po persistenční vrstvu,
- logická struktura infrastrukturních tříd frameworku umožňuje snadné přizpůsobení jejich chování rozšířením,
- sjednocující princip definice objektů a jejich provázanosti v konfiguračním souboru, který umožňuje například i jednoduchou testovatelnost jednotek aplikace,
- nebývale vysoká míra pokrytí kódu samotného frameworku testy, což zaručuje jeho stabilitu.

## 5.6 Nevýhody

Bohužel Spring MVC má také své nevýhody, naštěstí je jich ve srovnání s výhodami znatelně méně.

- Konfigurace XML je poměrně složitá
- Složitější realizace MVC
- V základní verzi Spring MVC chybí podpora AJAX

## 5.7 Shrnutí

Spring MVC se nezabývá řešením již vyřešených problémů. Místo toho využívá prověřených a dobře fungujících existujících open-source nástrojů, které v sobě integruje. Tím se stává jejich použití často jednodušším. Spring je modulární framework a umožňuje využít třeba jen část, která se zrovna hodí k řešení daného problému. Jeho účelem je zjednodušení návrhu J2EE aplikací se zaměřením na architekturu aplikace místo na technologii, jednoduchou testovatelnost, neinvazivnost a modulárnost.

## 6 Vaadin

### 6.1 Úvod

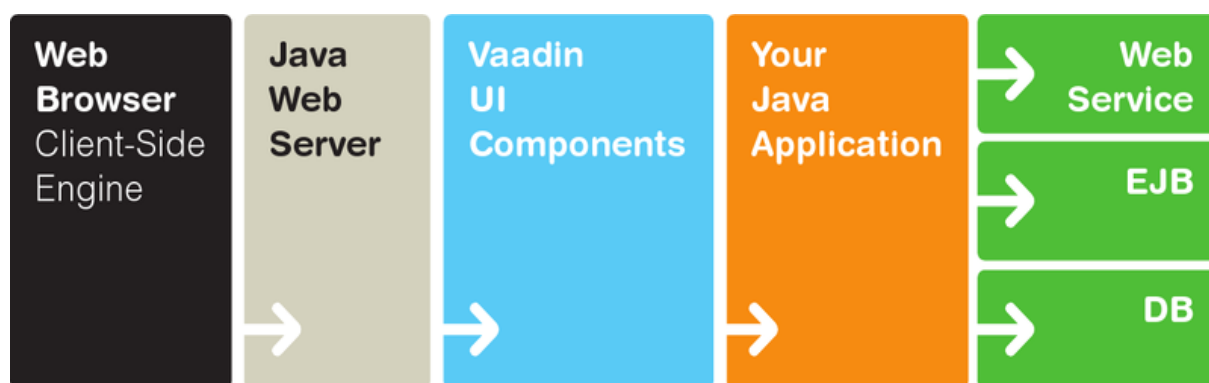
Vaadin [4] je softwarový framework pro tvorbu Javovských aplikací. Tyto aplikace se v internetovém prohlížeči chovají úplně stejně jako desktopové aplikace. Samotný kód je psán samozřejmě v Jave, ten je následně pomocí GWT překládán do JavaScriptu a ten teprve interpretován ve webovém prohlížeči.

### 6.2 Vlastnosti

Až na drobné výjimky programátor vůbec nepozná, že nepíše desktopovou aplikaci, je odstíněn od bezstavovosti protokolu HTTP. Programátor rovněž nepracuje přímo z HTML či JavaScriptem. Layoutu je docíleno obdobně jako u swingu pomocí panelů. JavaScript je generován pomocí GWT kompilátoru.

### 6.3 Architektura

Vaadin využívá GWT (Google Web Toolkit) [9], který slouží pro vykreslení stránky ve webovém prohlížeči. Tyto aplikace jsou napsány v Javě a poté přeloženy do JavaScriptu. GWT je vhodné pro tvorbu pokročilých komponent uživatelského rozhraní a implementaci logiky, která má být vykonávána v prohlížeči (v GWT terminologii se těmito komponentám říká widgetset). Na straně Vaadinu tedy spočívá správa aplikační logiky. Vaadin je navržen jako rozšiřitelný framework. Proto je možné využívat GWT komponenty třetích stran společně s Vaadin komponentami. Díky GWT lze všechen kód psát čistě v Javě. (Volný překlad z [10]).



Obrázek 3 - Vaadin architektura

## 6.4 Dokumentace

Co se týká dokumentace [4] je na tom Vaadin spolu se Spring MVC nejlépe z mnou porovnávaných webových frameworků. Vaadin navíc disponuje nepřeberným množstvím různých dem a funkčních příkladů [zdroj]. V tomto ohledu je ze všech mnou porovnávaných frameworků nejlepší a nejkvalitněji zpracovaný.

Vaadin disponuje na svém webu publikací Book of Vaadin [4]. V této 576 stránkové (velmi rozsáhlé) publikaci programátor nalezne vše potřebné. I zde je možnost dokument prohlížet jako HTML stránku nebo si jej stáhnout ve formátu PDF.

## 6.5 Výhody

- Obsahuje velkou sadu běžně používaných komponent pro tvorbu webových aplikací.
- K dispozici je seznam a online ukázky včetně zdrojových kódů.
- Součástí pluginu pro Eclipse je také vizuální editor pro tvorbu vlastních komponent.
- Komunita vývojářů přidává neustále nové komponenty a sdílí je s ostatními jako takzvané doplňky (add-ons).
- Využívá lazy loading (líné načítání)
- Podporuje drag & drop.

## 6.6 Nevýhody

- Není zde možnost využití tlačítka „Zpět“
- Rychlost

## 6.7 Shrnutí

Vaadin je ze všech mnou testovaných frameworků nejmladší, ale i přesto má širokou základnu příznivců, kteří se starají o jeho aktuálnost a podporu. Bezesporu největší výhodou a změnou oproti ostatním frameworkům je to, že programátor, jak již bylo řečeno, vůbec (až na výjimky) nepozná, že píše webovou aplikaci a ne klasickou desktopovou.

## 7 Struts2

### 7.1 Úvod

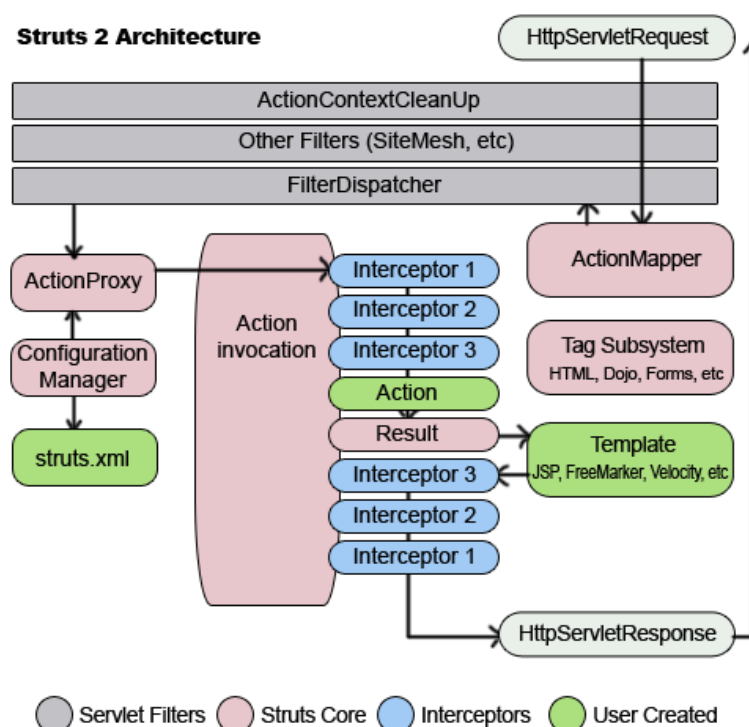
Framework Struts2 [3] je postaven přímo na J2EE. Struts2 se snaží oddělit od sebe vrstvu aplikační logiky od vrstvy prezentační. Je vyvíjen v rámci Apache Software Foundation pod licencí Apache Licence 2.0. Původní verze spatřila světlo světa v roce 2000. Díky této dlouhé historii mají Strutsy širokou komunitu. Řada Struts2 vznikla v roce 2005 a je do jisté míry odlišná od původní verze.

### 7.2 Vlastnosti

Dá se říct, že následníkem tohoto frameworku je Spring MVC, který se snaží vyřešit některé nedostatky tohoto frameworku. Stejně jako Spring MVC plně podporuje MVC architekturu, ta se však od svého předchůdce, tedy od frameworku Struts 1, výrazně liší. Spring 2 plně podporuje technologii AJAX. Je zde rovněž podpora validace dat z formulářů, ta je ovšem proti Struts1 méně robustní a je nutné definovat pravidla.

### 7.3 Architektura

Ve frameworku Struts2 je použit návrhový vzor Inversion of Control, díky tomuto vzoru lze za akci považovat všechny objekty POJO (Plain Old Java Object), které obsahují metodu `execute()` a vracejí datový typ `String`. Framework Struts2 obsahuje rovněž řadu rozhraní pro testování a rozšiřování aplikace. Tím je zajištěna určitá flexibilita.



Obrázek 4 - Struts2 architektura

## 7.4 Dokumentace

V dokumentaci Struts2 [11] jsem ze začátku vývoje testovací aplikace trochu tápal, ale nakonec jsem i zde našel vše potřebné. Všeobecně známým problémem je snadná zaměnitelnost s předchůdcem Struts2, tedy Struts1. Je dobré tedy hledat weby, které se věnují výhradně nové verzi Struts2 ([www.roseindia.net](http://www.roseindia.net), [www.struts2.org](http://www.struts2.org)). V dokumentaci rovněž není úplně jednoduchá orientace, ale naproti tomu obsahuje řadu ukázek a návodů, což značně usnadňuje začátky s tímto frameworkem.

## 7.5 Výhody

- Dobře spolupracuje s frameworkem Spring
- Neskryvá bezstavovost http protokolu
- Podporuje stavovost některých prvků

## 7.6 Nevýhody

- Nepřináší žádné inovace v programovém stylu

- Chybí rozšířené možnosti pro uživatelské rozhraní
- Přílišná složitost XML souborů

## 7.7 Shrnutí

Struts2 je framework, který pozvednul myšlenku původního frameworku Struts1 a přinesl modernější vývoj webových aplikací. Struts2 rovněž obsahuje velké množství technologií a vytváří velice robustní variantu pro webový vývoj. Ačkoliv jde o následovníka původního frameworku Struts 1, má s ním mnoho společného, lze tedy jednoduše migrovat a je zajištěna zpětná kompatibilita.

## 8 Java Server Faces

### 8.1 Úvod

Základem Java Server Faces [2] je technologie JSP (Java Server Pages) [2], ve které funguje systém knihoven tagů. Díky těmto tagům může programátor vkládat na JSP stránky komponenty uživatelského rozhraní. (UI komponenty). Pro využití Java Server Face programátor vkládá na stránky předpřipravené komponenty a mapuje je na příslušné třídy, které těmto komponentám předávají funkcionalitu.

### 8.2 Vlastnosti

Java Server Faces je komponentově orientovaný webový framework, je tedy jiného typu než předchozí Struts2. V klasickém pojetí je stránka obsluhována jako celek, je zde velký důraz na usnadnění obsluhy paradigmatu HTTP Request-Response. Komponentový přístup JSF deleguje obsluhu stránky jako takové na jednotlivé komponenty, které si svoje místo na webové stránce obhospodařují samy.

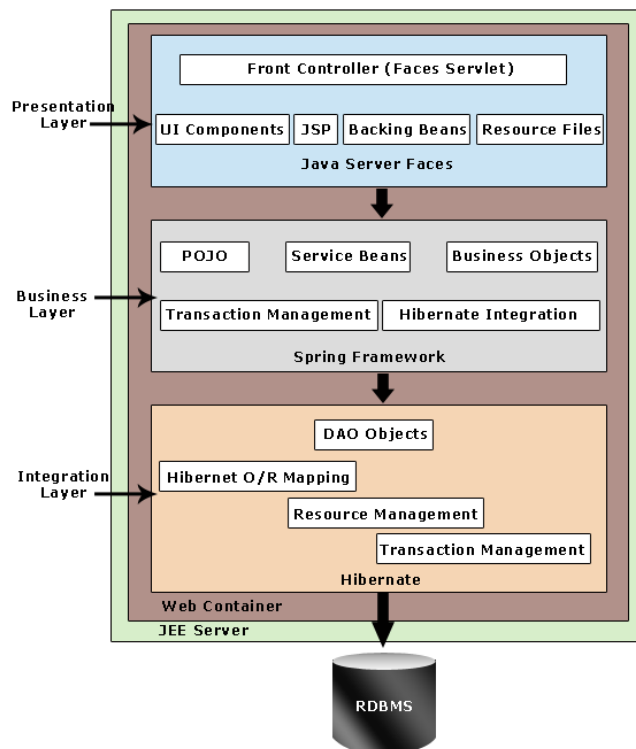
JSF, stejně jako většina webových frameworků, je postaveno na Servlet API a JSP [12]. JSP jako view vrstva může být ovšem nahrazena prakticky čímkoliv. Pro JSP jsou k dispozici dvě knihovny značek (taglib).

Jednou z nejčastěji kritizovaných vlastností JSF je obecnost, se kterou lze v JSF vyměnit prakticky kteroukoliv podčást za jinou. Ve spojení s jednoduchostí použití (o což se JSF snaží) je pod povrchem JSF schováno hodně kódu, který je těžké pochopit a při chybě těžké opravit.

### 8.3 Architektura

Java Server Faces jsou, jak již bylo řečeno, založeny na architektuře Model View Controller (MVC) [6], což znamená důsledné oddělení stavu, logiky a prezentace. Toho však lze dosáhnout za použití Java Server Pages (JSP) [12].





Obrázek 5 - JSF architektura

## 8.4 Dokumentace

Řekl bych, že dokumentace JSF [2] je celkově na dobré úrovni. Hlavní výhodu vidím v tom, kolik programátorů se kolem Java Server Faces pohybuje, proto není problém najít hledanou informaci, či kvalitní ukázkou a podobně. Samotná dokumentace je rozumně členěná a celá na jednom místě, osobně jsem neměl problém v ní najít vše, co jsem potřeboval.

## 8.5 Výhody

- Vkládání UI komponent na JSP [12] stránky pomocí dostupných knihoven tagů
- Linkování komponent, které generují události, na Javovký kód, který tyto události zpracovává
- Validace a konverze uživatelských vstupů.
- Tvorba znovupoužitelných komponent.
- Ukládání a načítání stavu komponent mezi požadavky.

## 8.6 Nevýhody

- Příliš komplexní řešení
- Komplikovaný debugging
- Chybí možnost kombinovat hlavní sady komponent

## 8.7 Shrnutí

Framework Java Server Faces, umožňuje vytvářet webové aplikace způsobem, který je velmi podobný vývoji desktopových aplikací. JSF se většinou označuje jako komponentový, ačkoliv podle mě to nelze takhle paušálně říci a takto může být nazývána jen některá z jeho konkrétních implementací. Hlavní myšlenka je v znovu použití UI komponent, jak již bylo rovněž řečeno, se jedná o vhodný framework pro vývojáře desktopových aplikací.

## 9 Testovací aplikace

### 9.1 Úvod

Jako testování (srovnávací) aplikaci implementovanou ve všech mnou vybraných frameworkcích jsem zvolil jednoduchý e-shop.

### 9.2 Popis testovací aplikace

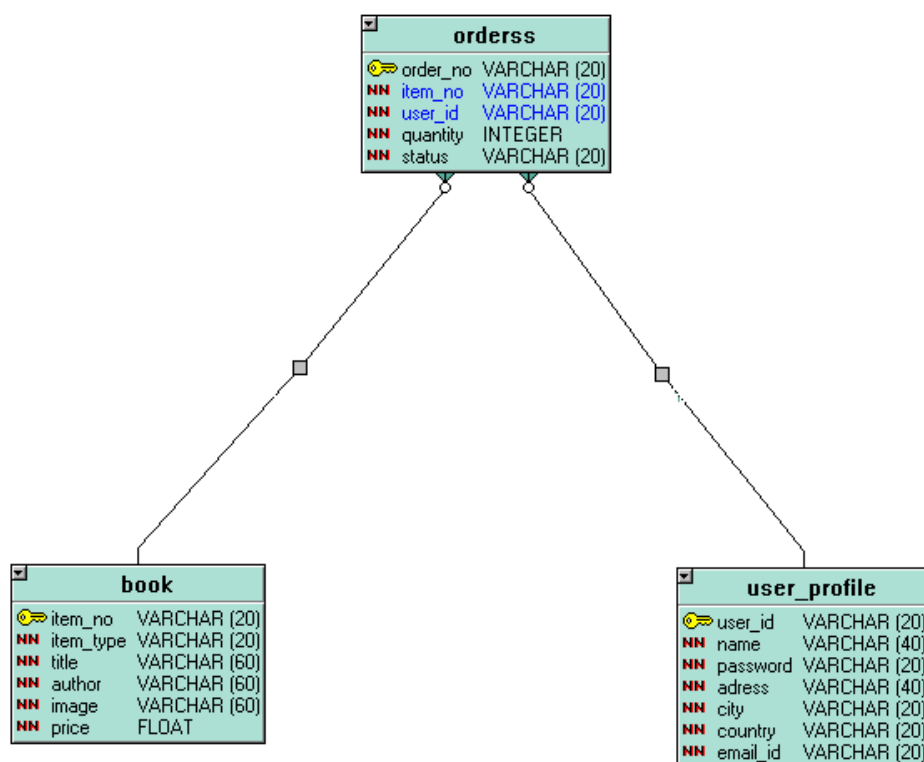
Jako druh zboží pro e-shop jsem zvolil knihy. Aplikace je postavena na všech čtyřech frameworkcích. Pod jednotlivými frameworky běží databázový Framework Hibernate, který slouží pro přístup do databáze. Schéma databáze je uvedeno níže. Jak již bylo řečeno, všechna data jsou uložena v databázi, a to konkrétně v databázi MySQL.

Aplikace funguje jako standartní e-shop, zboží je možno prohlížet a listovat jím, ale pro objednání zboží (přidání do košíku) je nutno se přihlásit. Údaje jsou ověřovány naproti databázi. Pro přihlášení do aplikace je nutné se nejdříve registrovat v aplikaci a vyplnit příslušné údaje. Po registraci se již uživatel může standardně přihlásit. Po přihlášení se mu objeví možnost zboží přidat do košíku, editovat košík a ukončit (potvrdit) objednávku.

Aplikace rovněž obsahuje jednoduchou administrátorskou část, ve které může administrátor, přidávat nebo odebírat zboží s e-shopu. Z tohoto je patrné, že e-shop má klasický dvou úroňový systém rolí- Administrátor, Zákazník.

## 9.2.1 Schéma databáze

PROJEKT:	E-SHOP
VERZE:	1.0
DB SERVER:	InterBase / Firebird
AUTOR:	Jakub Ribka
FIRMA:	VŠB
VYTVOŘENO:	27.04.2013 20:27
UPRAVENO:	27.04.2013 20:47



Obrázek 6 - Schéma databáze

## 9.2.2 Tabulky

Jedná se sice o e-shop, ale jen v zjednodušené verzi a proto, kdyby bylo uvažováno o ostrém nasazení tohoto e-shopu, byla by potřeba spousta úprav a schéma databáze by se určitě lišilo od našeho testovacího příkladu. Nám ale toto schéma a tato aplikace bohatě postačí a můžeme na ní pohodlně testovat a srovnávat řadu kritérií daných frameworků.

### 9.2.2.1 Book

Tabulka **book** reprezentuje tabulku zboží v našem případě tedy knih, které zákazníci nakupují. Je reprezentována několika sloupci, z nichž nejdůležitější si popíšeme. Hlavní je primární klíč, jelikož se bavíme o knihách, je to ISBN a v tabulce má název *item\_no*. Další zajímavým políčkem je *item\_type* a toto udává, o jakou knihu se jedná např. technika, náboženství, učebnice atd. Ostatní políčka již jsou celkem samozřejmá a nepotřebují dalšího vysvětlování.

```
create table book_shop(
item_no varchar(20) not null,
item_type varchar(20) not null,
title varchar(60) not null,
author varchar(60) not null,
about varchar(10000) not null,
image varchar(60) not null,
price float not null,
primary key(item_no));
```

### 9.2.2.2 Orders

Tabulka **orders**, neboli objednávky, je hlavní tabulkou, která spojuje tabulky **book** (kniha) a **user\_profile** (zákazník). Vazba je patrná ze schématu databáze, které je umístěno výše. Primární klíč této tabulky je číslo objednávky, což by v reálu nemuselo být úplně ideální, ale pro náš testovací případ je to naprosto dostačující.

```
create table orderess(
order_no varchar(20)not null ,
user_id varchar(20) not null,
item_no varchar(20) not null,
quantity int not null default 0,
status varchar(20) not null,
primary key(order_no));
```

### 9.2.2.3 User\_profile

Tabulka **user\_profile** (zákazník) slouží k přihlašování a evidenci zákazníků. Její primární klíč je **user\_id**, které zároveň slouží jako login při přihlašování do aplikace. Jak vidíte, původní návrh má spoustu políček, tak aby mohl být použit ve skutečném e-shopu. Nicméně my tolik políček využívat nebudeme a některá zbytečná vypustíme.

```
create table user_profile(
name varchar(40) not null,
user_id varchar(20) not null,
password varchar(20) not null,
address_line1 varchar(40) not null,
address_line2 varchar(40) default null,
city varchar(20) not null,
country varchar(20) not null,
pin varchar(20) not null,
gender varchar(20) not null,
age varchar(20) not null,
```

```
email_id varchar(20) not null,  
phone_number varchar(20) not null,  
card_no varchar(20) not null,  
expiry_date varchar(20) not null,  
car_type varchar(20) not null,  
account_balace float not null,  
primary key(user_id));
```

## 10 Grafické ukázky v jednotlivých frameworkcích

Aplikace jsou, co se funkčnosti týče, naprosto totožné. Po designové stránce byla snaha o co největší podobnost a shodu jednotlivých aplikací. Na ukázky implementované v jednotlivých frameworkcích se můžete podívat v níže přiložených screenech. Ukážeme si úvodní stránku aplikace, registraci nového zákazníka a košík zboží.

### 10.1 Vaadin

#### 10.1.1 Úvodní strana



Obrázek 7 - Úvodní strana, Vaadin

## 10.1.2 Košík

The screenshot shows the Vaadin website's shopping cart interface. At the top, there's a navigation bar with the Vaadin logo and a search box. Below it, a menu contains links: ÚVOD, O NÁS, OBCHODNÍ PODMINKY, DOPRAVA, REGISTRAČNÍ FORMULÁŘ, and KONTAKT. The main content area is divided into two columns. The left column, titled 'Přihlášen', shows the user 'Jakub Ribka' with email 'r.jakub@seznam.cz' and buttons for 'Odhlásit' and 'Zobrazit košík'. Below this is a 'Knihy' section with a list of book categories: Proza, Umění, Příroda, Učebnice, Technika, Medicína, Historie, Zvířata, Sport, and Maboženství. The right column, titled 'Následující položky jsou ve Vašem košíku', displays a table with one item: 'Košíkaření z papíru' (category: umeni), quantity 1, price 600.0 Kč. Below the table, it shows 'Celková cena objednávky: 600.0 Kč' and a 'Pokračovat v objednávce' button. At the bottom, there's a 'Kontaktujte nás' section with the address 'Adresa: Ostrava, Krasná 3'.

Číslo	Název	Druh položky	Množství	Cena
1	Košíkaření z papíru	umeni	1	600.0 Kč

Obrázek 8, Nákupní košík, Vaadin

## 10.1.3 Registrace

The screenshot shows the Vaadin website's registration form. It features the same navigation bar and menu as the shopping cart page. The main content area is divided into two columns. The left column, titled 'Přihlášení', contains fields for 'Username' and 'Password', a 'Login' button, and the same 'Knihy' category list. The right column, titled 'Registrace', contains a series of input fields for registration: 'Jméno, Příjmení', 'Uživatel', 'Heslo', 'Potvrdit heslo', 'Adresa 1', 'Adresa 2', 'Město', 'Stát', 'PSČ', 'Pohlaví' (a dropdown menu), 'Věk', 'E-mail', and 'Telefon'. Each field has a red asterisk indicating it is required. A 'Registrovat' button is located at the bottom of the registration fields. The 'Kontaktujte nás' section at the bottom remains the same, showing the address 'Adresa: Ostrava, Krasná 3'.

Obrázek 9 - Registrace, Vaadin



## 10.2 Java Server Faces

### 10.2.1 Úvodní strana

**JavaServer™ Faces JSF**

Úvod O nás Obchodní podmínky Doprava Registrační formulář Kontakt

Náš tým doporučuje tyto knihy za akční ceny:

**Přihlášení**  
Jméno  
user  
Heslo  
\*\*\*\*  
Přihlásit

**Knihy**


- Umění
- Příroda
- Učebnice
- Technika
- Medicína
- Historie
- Zvířata
- Sport
- Náboženství
- Próza

**Kontaktujte nás**  
Adresa: Ostrava, Krasná 3  
Adresa: Ostrava, Krasná 3  
Adresa: Ostrava, Krasná 3  
Kontakt: 608034123.

Knihy	Cena
Dějiny filmu	900.0 Kč
Kameny domova	400.0 Kč
Medical English	1110.0 Kč
Kameny domova	400.0 Kč
Projekty výchovy	1300.0 Kč
Projekty výchovy	1300.0 Kč
Dějiny filmu	malý
Projekty výchovy	malý
Obrázky z chatrče	malý

Obrázek 10 - Úvodní strana, JSF

## 10.2.2 Registrace



[Úvod](#) [O nás](#) [Obchodní podmínky](#) [Doprava](#) [Registrační formulář](#) [Kontakt](#)

**Registrace:**

**Přihlášení**  
Jméno   
Heslo

**Knihy**  

Umění  
Příroda  
Učebnice  
Technika  
Medicína  
Historie  
Zvřřata  
Sport  
Náboženství  
Próza

**Registrace:**  
Jméno   
Příjmení   
Email   
Login   
Heslo

[Kontaktujte nás](#)  
Adresa: Ostrava, Krasná 3  
Adresa: Ostrava, Krasná 3  
Adresa: Ostrava, Krasná 3  
Kontakt: 608034123.

Obrázek 11 - Registrace, JSF

## 10.2.3 Košík



[Úvod](#) [O nás](#) [Obchodní podmínky](#) [Doprava](#) [Registrační formulář](#) [Kontakt](#)

**Následující položky jsou ve Vašem košíku:**

**Přihlášení**  
user Odhlásit  
**Košík**  
**Knihy**  

Próza  
Umění  
Příroda  
Učebnice  
Technika  
Medicína  
Historie  
Zvřřata  
Sport  
Náboženství

Číslo	Název	Druh položky	Množství	Cena
00016aaf-f9a1-46f2	Košíkaření z papíru	umeni	1	600.0

**Celková cena objednávky 600.0 Kč**  
[Pokračovat v objednávce](#)

[Kontaktujte nás](#)  
Adresa: Ostrava, Krasná 3  
Adresa: Ostrava, Krasná 3  
Adresa: Ostrava, Krasná 3  
Kontakt: 608034123.

Obrázek 12 - Košík, JSF

## 10.3 Struts2

### 10.3.1 Úvodní strana

# Struts<sup>2</sup>

[Úvod](#) [O nás](#) [Obchodní podmínky](#) [Doprava](#) [Registrační formulář](#) [Kontakt](#)

Náš tým doporučuje tyto knihy za akční ceny:

Přihlášení

Přihlášení

Knihy

Umění  
Příroda  
Technika  
Učebnice  
Medicína  
Historie  
Zvířata  
Sport  
Naboženství  
Proza

Kontaktujte nás  
Adresa: Ostrava, Krasná 3  
Adresa: Ostrava, Krasná 3  
Adresa: Ostrava, Krasná 3  
Kontakt: 608034123.

<div>Dějiny filmu</div> <div></div> <div>900.0 Kč</div>	<div>Kameny domova</div> <div></div> <div>400.0 Kč</div>	<div>Medical English</div> <div></div> <div>1110.0 Kč</div>
<div>Kameny domova</div> <div></div> <div>400.0 Kč</div>	<div>Projekty výchovy</div> <div></div> <div>1300.0 Kč</div>	<div>Projekty výchovy</div> <div></div> <div>1300.0 Kč</div>
<div>Dějiny filmu</div> <div></div>	<div>Projekty výchovy</div> <div></div>	<div>Obrázky z chatrče</div> <div></div>

Obrázek 13 - Úvodní strana, Struts2

### 10.3.2 Registrace

**Struts<sup>2</sup>**

Úvod O nás Obchodní podmínky Doprava Registrační formulář Kontakt

Přihlášení

user

\*\*\*\*

Přihlášení

Knihy

Umění  
Příroda  
Technika  
Učebnice  
Medicína  
Historie  
Zvířata  
Sport  
Náboženství  
Proza

Kontaktujte nás  
Adresa: Ostrava, Krasná 3  
Adresa: Ostrava, Krasná 3  
Adresa: Ostrava, Krasná 3  
Kontakt: 608034123

Náš tým doporučuje tyto knihy za akční ceny:

Jméno

Příjmení

Email

Login

Heslo

Registrace

Obrázek 14 - Registrace, Struts2

### 10.3.3 Košík

**Struts<sup>2</sup>**

Úvod O nás Obchodní podmínky Doprava Registrační formulář Kontakt

Přihlášen

Jakub Ribka

r.jakub@seznam.cz

Košík

Odhlásit

Knihy

Proza  
Umění  
Příroda  
Učebnice  
Technika  
Medicína  
Historie  
Zvířata  
Sport  
Náboženství

Kontaktujte nás  
Adresa: Ostrava, Krasná 3  
Adresa: Ostrava, Krasná 3

Následující položky jsou ve Vašem košíku:

Číslo	Název	Druh	Množství	Cena	
Číslo	Košíkaření z papíru	Druh	umění	1	600.0

Celková cena objednávky je 600.0

Pokračovat v objednávce

Obrázek 15 - Košík, Struts2

## 10.4 Spring MVC

### 10.4.1 Úvodní strana

The screenshot displays the homepage of the Spring MVC application. At the top is a navigation bar with links: Úvod, O nás, Obchodní podmínky, Doprava, Registrační formulář, and Kontakt. Below this, a banner states: "Náš tým doporučuje tyto knihy za akční ceny:". The main content area is divided into a sidebar on the left and a grid of book cards on the right. The sidebar includes a "Přihlášení" (Login) section with input fields and a "Přihlášení" button, and a "Knihy" (Books) section with a list of categories: Próza, Umění, Příroda, Učebnice, Technika, Medicína, Historie, Zvířata, Sport, and Náboženství. Below the categories is a "Kontaktujte nás" (Contact us) section with the address "Adresa: Ostrava, Krasná 3" and contact number "Kontakt: 608034123". The grid of book cards features six books, each with a title, cover image, price (600.0 Kč), and a shopping basket icon. The books are: "Košíkaření z papíru" (Paper Basket Weaving), "Dějiny filmu" (History of Film), "Kameny domova" (Home Stones), "Medical English", "Kameny domova" (Home Stones), and "Projekty výchovy" (Education Projects). The book covers show various themes, including film history, language learning, and traditional crafts.

**spring**

Úvod O nás Obchodní podmínky Doprava Registrační formulář Kontakt

Náš tým doporučuje tyto knihy za akční ceny:

**Přihlášení**

**Knihy**

Próza  
Umění  
Příroda  
Učebnice  
Technika  
Medicína  
Historie  
Zvířata  
Sport  
Náboženství

Kontaktujte nás  
Adresa: Ostrava, Krasná 3  
Adresa: Ostrava, Krasná 3  
Adresa: Ostrava, Krasná 3  
Kontakt: 608034123.

**Košíkaření z papíru**  
Dějiny filmu  
Kameny domova

**angličtina pro jazykové školy**

**Medical English**  
Kameny domova  
Projekty výchovy

**Václav Čížek dýchat s ptáky**  
Košíkaření z papíru  
BIOCHEMIE PRO STUDIJÍ MEDICÍNY I. díl

600.0 Kč 600.0 Kč 600.0 Kč 600.0 Kč 600.0 Kč 600.0 Kč

Projekty výchovy Dějiny filmu Projekty výchovy

Obrázek 16 - Úvodní strana, Spring

## 10.4.2 Nákupní košík



<a href="#">Úvod</a>	<a href="#">O nás</a>	<a href="#">Obchodní podmínky</a>	<a href="#">Doprava</a>	<a href="#">Registrační formulář</a>	<a href="#">Kontakt</a>
----------------------	-----------------------	-----------------------------------	-------------------------	--------------------------------------	-------------------------

Následující položky jsou ve Vašem košíku:

Přihlášen	Číslo	Název	Druh	Množství	Cena
<b>Jméno</b> Jakub Ribka	Číslo	Dějiny filmu	umení	1	900.0
<b>Login</b> user	Číslo	Kameny domova	ucebnice	1	400.0
<b>E-mail</b> r.jakub@seznam.cz	Celková cena objednávky je 1300.0				
<a href="#">Pokračovat v objednávce</a>					


**Knihy**

- Próza
- Umění
- Příroda
- Učebnice
- Technika
- Medicína
- Historie
- Zvířata
- Sport
- Náboženství

[Kontaktujte nás](#)  
Adresa: Ostrava, Krasná 3

Obrázek 17 - Nákupní košík, Spring

## 10.4.3 Registrace



<a href="#">Úvod</a>	<a href="#">O nás</a>	<a href="#">Obchodní podmínky</a>	<a href="#">Doprava</a>	<a href="#">Registrační formulář</a>	<a href="#">Kontakt</a>
----------------------	-----------------------	-----------------------------------	-------------------------	--------------------------------------	-------------------------

Registrace:

<b>Přihlášení</b>	<b>Jméno</b>
Login: <input type="text"/>	<input type="text"/>
Heslo: <input type="text"/>	<b>Příjmení</b>
<input type="button" value="Submit"/>	<input type="text"/>
	<b>Email</b>
	<input type="text"/>
	<b>Login</b>
	<input type="text"/>
	<b>Heslo</b>
	<input type="text"/>
	<input type="button" value="Registrace"/>

**Knihy**

- Próza
- Umění
- Příroda
- Učebnice
- Technika
- Medicína
- Historie
- Zvířata
- Sport
- Náboženství

[Kontaktujte nás](#)  
Adresa: Ostrava, Krasná 3  
Adresa: Ostrava, Krasná 3  
Adresa: Ostrava, Krasná 3  
Kontakt: 608034123.

Obrázek 18 - Registrace, Spring

## 11 Srovnání vybraných frameworků

Nyní si rozebereme vývojové etapy v jednotlivých frameworkcích, zaměříme se na rychlost vývoje, dostupnou dokumentaci, ale hlavně na výkon výsledné aplikace na základě zátěžových a výkonnostních testů. Podrobněji si rozebereme některé základní prvky webových aplikací, které se standardně vyskytují ve většině webových aplikací. Na konci každé kapitoly uvedu tabulku, kde bude ohodnoceno body od 1 do 10, jak si daný framework stojí v té či oné kapitole.

### 11.1 Instalace pluginů

Instalace všech frameworků (Java Server Faces, Struts2, Vaadin, Spring MVC) probíhá stejně. Stažením pluginu (\*.JAR soubor), který je následně přilinkován k webovému projektu.

#### 11.1.1 Velikosti jednotlivých pluginů

Frameworky mohou být distribuovány v mnohá podobách a verzích. Pro naši aplikaci byly zvoleny základní pluginy, bez ukázkových aplikací či jiných rozšiřujících struktur. Základní velikost porovnávaných frameworků:

- Vaadin 4,82 MB
- Struts2 3,8 MB
- Spring MVC 5,2 MB
- Java Server Faces 1,8 MB

## 11.2 Složitost a rychlost vývoje jednotlivých prvků aplikace

### 11.2.1 Vaadin

Vývoj testovací aplikace v tomto frameworku pro mě byla ze všech čtyř testovaných frameworků nejjednodušší a nejrychlejší, jelikož jsem pracoval rok ve firmě Tieto, kde jsem právě s tímto frameworkem pracoval. Nicméně pokusím se nastínit jeho složitost a rychlost vývoje pro uživatele, který s ním nemá větší zkušenosti.

Největší změnou pro programátora, který je zvyklý na klasické jsp či html (xhtml) stránky je to, že aplikace ve Vaadinu se píše jako klasická desktopová aplikace a programátor vůbec nepříjde do styku s view stránkami. Vývoj jednotlivých komponent je ovšem díky kvalitním demům a ukázkám velmi rychlý a efektivní. Vaadin obsahuje všechny standartní prvky, které webové aplikace využívají, navíc obsahuje spoustu komponent, které další mnou srovnávané frameworky nemají, nebo se s nimi obtížně pracuje.



### 11.2.2 Java Server Faces

Technologii JSF jsem rovněž znal, tentokrát ze školních projektů, takže nebyl velký problém s rychlostí vývoje. Ale pokud jde o Java Server Faces pro programátora, který se s nimi setkává poprvé, jsou na tom, podle mého názoru, co se rychlosti a složitosti vývoje týká, ze všech mnou porovnávaných frameworků nejlépe. Technologie JSF je přehledná, vývoj je velmi rychlý, a to hlavně díky široké internetové komunitě zabývající se tímto frameworkem a jednak tím, že tato technologie má už nějaký ten pátek své místo mezi ověřenými webovými frameworky.

### 11.2.3 Spring MVC

Framework Spring MVC byl pro mě úplně novým frameworkem, pod kterým jsem ještě žádný projekt nevyvíjel. Nicméně po prvotním seznámení a nastudování (ke kterému se ještě vrátím v následující kapitole) jde vývoj s frameworkem Spring MVC dobře od ruky. Narazil jsem sice na určité komplikace, na které jsem nebyl připravený, například jsem měl problém s přihlašovaním do aplikace a se správou session. Tyto problémy způsobily, že vývoj testovací aplikace pod tímto frameworkem byl druhý nejdelší. Nicméně po vyřešení všech problémů, na které jsem narazil, mohu prohlásit, že Spring MVC je účinný nástroj pro vývoj webových aplikací na platformě Java.

### 11.2.4 Struts2

Struts2 byl další z frameworků, který byl pro mě novinkou, a bylo nutno si jej nejprve osahat. Největším problémem pro mě zpočátku bylo složité konfigurování XML souboru pro jednotlivé Actions. Toto byl asi hlavní důvod, proč se framework Struts2 umístil v mém vyhodnocení, co se rychlosti a složitosti vývoje týká, na posledním místě, a to za výše zmiňovaným frameworkem Spring MVC. Ale opět bych řekl, že hlavní problém byl ve zvyku z jiných frameworků, jakmile si programátor zvykne na konvenci a strukturu definující např. Actions a podobné specifiky frameworku Struts2, dá se vývoj v něm nazvat efektivním.

### 11.2.5 Vyhodnocení

Podle níže uvedené tabulky vidíme, že kdyby naším kritériem byla rychlost a složitost vývoje, tak nejlépe se prezentuje Java Server Face, jelikož v nich vývoj jde opravdu „rychle od ruky“, naopak nejhůře dopadly Struts2, dle mého názoru je to dáno složitým používáním XML souborů.

Činnost/Framework	Vaadin	JSF	Spring MVC	Struts2
Rychlost vývoje	8	9	5	6
Složitost vývoje	6	6	7	5
<b>CELKEM (avg.)</b>	<b>7</b>	<b>8</b>	<b>6</b>	<b>6</b>

Tabulka 1 - Složitost a rychlost vývoje



## 11.3 Náročnost prvotního seznámení s technologií

Podíváme se na jednotlivé frameworky z pohledu prvotního seznámení s danou technologií. Nastíním zde rovněž záludnosti, se kterými jsem se setkal při studování jednotlivých frameworků.

### 11.3.1 Vaadin

Jak již bylo řečeno, technologii Vaadin jsem již měl nastudovanou z dřívější doby, a bylo pouze nutné si jí, jak se říká, oprášit. Zkusíme se tedy podívat na tuto kapitolu ze strany Vaadinem nepolíbeného programátora. Vaadin má oproti ostatním třem frameworkům nevýhodu v této kategorii srovnání v tom, že se jedná opravdu o rozdílný framework, kde uživatel píše de facto desktopovou aplikaci a je odstíněn od view kódu. Toto hraje dle mého názoru zásadní roli při seznamování se s Vaadinem a proto se taky umístil v mém hodnocení na posledním místě. Je nutno ale podotknout, že za Vaadinem stojí výborná dokumentace s obsáhlými demy a ukázkami, které prvotní seznámení s technologií značně usnadňují, na tu se ale podíváme až v následující kapitole.

### 11.3.2 Java Server Faces

Myslím si, že pokud se chce programátor pustit to vývoje webové aplikace pod jedním z mnoha testovaných frameworků a nemá ani s jedním z nich zkušenosti, jsou pro něj Java Server Faces (aspoň co se prvotního seznámení týká) jako dělané. Jejich psaní je velmi intuitivní a proto počáteční studium a čas věnovaný seznámení se s technologií je velmi krátký. Tento fakt potvrzuje jednak už několikrát zmíněná obrovská komunita na internetu, ale také množství kvalitních publikací a spousty ukázek na internetu. Všechny tyto fakty vedou k tomu, že Java Server Faces se umístili v mém ohodnocení (prvotního seznámení s technologií) na prvním místě.

### 11.3.3 Spring MVC

Seznámení se s technologií Spring MVC by rovněž neměl být velký problém, jelikož se jedná v poslední době o velmi preferovaný a používaný framework, tolik aspoň suchá data. Pravdou ale zůstává, že v mém případě si framework Spring MVC vyžádal dlouhý čas strávený „zkoumáním a bádáním“, než jsem pochopil přesnou funkci tohoto frameworku. Zde je opět potřeba uvést velmi kvalitní dokumentaci, která seznamování se s tímto frameworkem usnadňuje, dle mého názoru je srovnatelná s frameworkem Vaadin, ale o té až níže. Bohužel pro mě jako pro programátora, který se s technologií Spring MVC ještě nesetkal, bylo prvotní seznámení celkem zdlouhavé, a proto jsem tento framework zařadil na předposlední místo z hlediska prvotního seznámení se s danou technologií.

### 11.3.4 Struts2

Technologii Struts2 bychom mohli nazvat jako předchůdce Spring MVC. Jedná se opět o technologii, se kterou jsem se doposud nesetkal. Začalo tedy mé studium a shánění informací o této technologii. Tato technologie je do jisté míry specifická svým složitým zápisem v XML souborech, na což jsem nebyl zvyklý. Proto bylo prvotní seznámení s touto technologií časově nejnáročnější ze všech

čtyř frameworků. Hlavním důvodem podle mého názoru je menší členská základna než u předešlých frameworků a z toho plynoucí menší počet příkladů a publikací, které usnadňují studium tohoto frameworku.

### 11.3.5 Vyhodnocení

Jak vidíme, nejvíce času zabírá seznámení s Vaadinem, je to dáno odlišným vývojem, než jsou programátoři u webových aplikací zvyklí. Nejlépe je na tom framework Java Server Faces, a to díky tomu, že aplikace je jednoduchá a přehledná, ale také díky široké komunitě, která se tomuto frameworku už léta věnuje.

Činnost/Framework	Vaadin	JSF	Spring MVC	Struts2
Seznámení s technologií	5	8	7	6
<b>CELKEM</b>	<b>5</b>	<b>8</b>	<b>7</b>	<b>6</b>

Tabulka 2 - Náročnost prvotního seznámení s technologií

## 11.4 Dostupnost a přehlednost dokumentace

### 11.4.1 Vaadin

Vaadin a jeho dokumentace [zdroj] je asi to nejkvalitnější, s čím jsme se u dokumentací webových frameworků setkal. Dokumentace je přehledně členěná a není problém v ní během okamžiku najít potřebné informace. Vaadin je z naší čtveřice nejmladší framework, lze tedy předpokládat, že dokumentace se bude vyvíjet bouřlivěji než u ostatní porovnávaných frameworků. Největší předností dokumentace Vaadinu jsou jeho velmi propracované demo a ukázkové příklady [zdroj]. Lze v nich najít příklady téměř na všechny komponenty, které Vaadin obsahuje a další přibývají.

### 11.4.2 Java Server Faces

Kvalita dokumentace frameworku Java Server Faces je dobrá a je dostupná na stránkách oraclu [zdroj]. V zásadě nebyl problém nalézt potřebné informace přímo v dokumentaci, která je přehledná a rozumně členěná. Pokud jsem měl nějaký problém s dokumentací (což se mi také stalo), nebylo složité najít problém na webu, jelikož, jak již bylo řečeno, velikost komunity u Java Server Faces je velká a jedná se o framework s dlouhou historií.

### 11.4.3 Spring MVC

Dokumentace frameworku Spring MVC [ydroj] je více méně srovnatelná s dokumentací Vaadinu. V čem však ovšem Spring za Vaadinem v dokumentaci výrazně pokulhává, jsou propracované dema. Jinak je dokumentace velmi přehledná a rozumně členěná. Lze ji stáhnout ve formátu PDF nebo přímo online ve formátu HTML [ydroj]. Nebyl problém v dokumentaci najít konkrétní odpovědi na dané otázky. Za tímto frameworkem stojí široká komunita programátorů a je v poslední době velmi oblíbený, což má na obsáhlost dokumentace jistě svůj vliv. Existuje rovněž celá řada odborných publikací, které se frameworkem zabývají.

### 11.4.4 Struts

Bohužel z hlediska dokumentace je na tom framework Struts2 s přehledem nejhůře. Dokumentace se mi zdála nepřehledná a špatně zpracována. Největší problém byla asi roztříštěnost dokumentace po různých souborech. Byl opravdový problém vyhledat odpověď na konkrétní otázku. Zjistil jsem, že předchůdce Struts 1 je na tom o poznání lépe, což mě vážně udivilo. Nicméně tento problém osobně přiřazuji menší členské základně, která se věnuje frameworku Struts2. Avšak hlavní problém je dle mého názoru ve frameworku Spring MVC, která s frameworku Struts vychází a snaží se odstranit jeho nedostatky, čímž částečně škodí i popularitě a tím pádem i velikosti komunity frameworku Struts2.

### 11.4.5 Vyhodnocení

Činnost/Framework	Vaadin	JSF	Spring MVC	Struts2
Seznámení s technologií	9	7	8	4
<b>CELKEM</b>	<b>9</b>	<b>7</b>	<b>8</b>	<b>4</b>

Tabulka 3 - Dostupnost a přehlednost dokumentace

## 11.5 Podpora v nástrojích pro vývoj

### 11.5.1 Vaadin

Vaadin byl bez problému použitelný ve vývojovém prostředí, ve které jsem pracoval tedy v Netbeansech. Stačilo pouze stáhnout plugin od Vaadin a ten přidat do Netbeansů a poté nebyl žádný problém začít s vývojem. Jak jsem zjistil, vydal Vaadin pro vývojové prostředí Eclipse, grafický editor výsledné stránky, něco jako když v netbeansech vytváříme desktopovou aplikaci. Nicméně tento plugin pro Eclipse jsem nestudoval.

## 11.5.2 Java Server Faces, Struts2, Spring MVC

Ve všech třech zbylých frameworkcích nebyl problém se stažením pluginu pro vývojové prostředí netbeans. Podporu jiných vývojových prostředí jsem nezkoumal. Vaadin jako jediný, který podporuje grafický návrh aplikace (v Eclipse), dostává v závěrečném hodnocení o bod více.

## 11.5.3 Vyhodnocení

Činnost/Framework	Vaadin	JSF	Spring MVC	Struts2
Seznámení s technologií	9	8	8	8
<b>CELKEM</b>	<b>9</b>	<b>8</b>	<b>8</b>	<b>8</b>

Tabulka 4 - Podpora v nástrojích pro vývoj

## 11.6 Výkonnost vzniklé aplikace na základě zátěžových a výkonnostních testů

Dostali jsme se k nejdůležitější části mé diplomové práce a hlavnímu měřítku srovnání všech vybraných frameworků a tím jsou výkonnostní a zátěžové testy.

### 11.6.1 Nástroje a nastavení

Testování proběhlo s těmito parametry:

#### A) Hardware

Windows edition			
Windows Server 2008 R2 Enterprise			
Copyright © 2009 Microsoft Corporation. All rights reserved.			
System			
Processor:	Intel(R) Xeon(R) CPU	X5670 @ 2.93GHz	2.92 GHz
Installed memory (RAM):	4,00 GB		
System type:	64-bit Operating System		

Obrázek 19 - Virtuální server

## B) Java EE server

Jako samotný server pro nasazení webové aplikace jsem zvolil Apache Tomcat 6. Jedná se o jeden z nejznámějších aplikačních serverů, který je založen na jazyku Java.

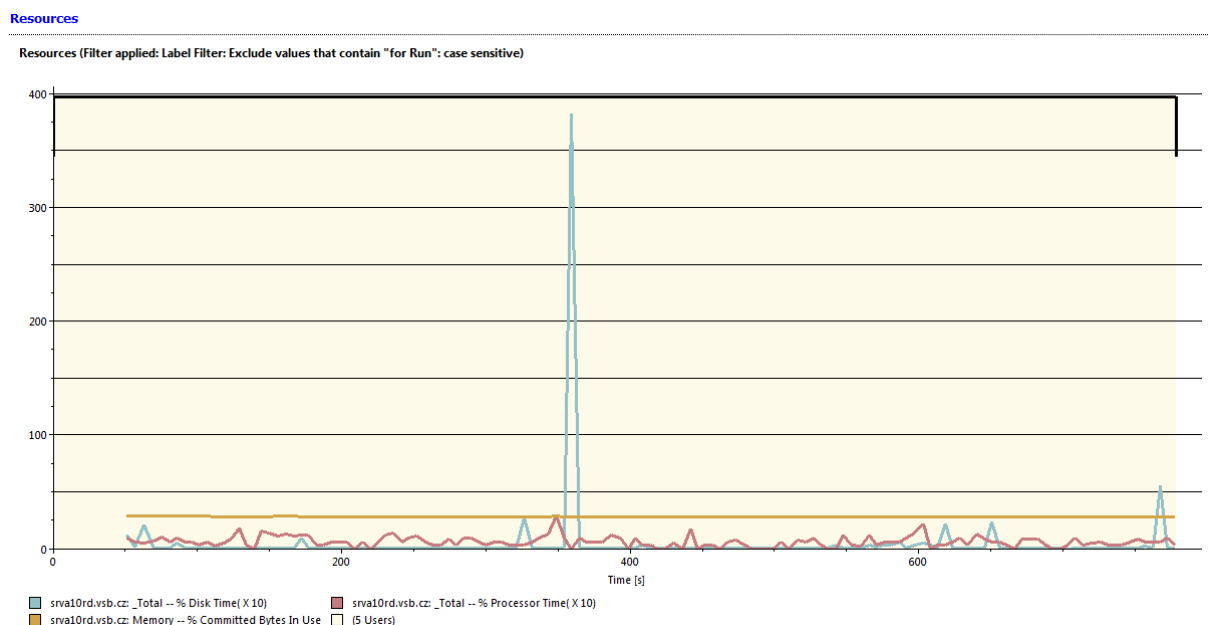
## C) Testovací nástroj

Testování proběhlo v IBM performance testeru. Produkt IBM Rational Performance Tester představuje řešení testování výkonu, jež ověřuje výkonnost webových a serverových aplikací.

## D) Uživatelé

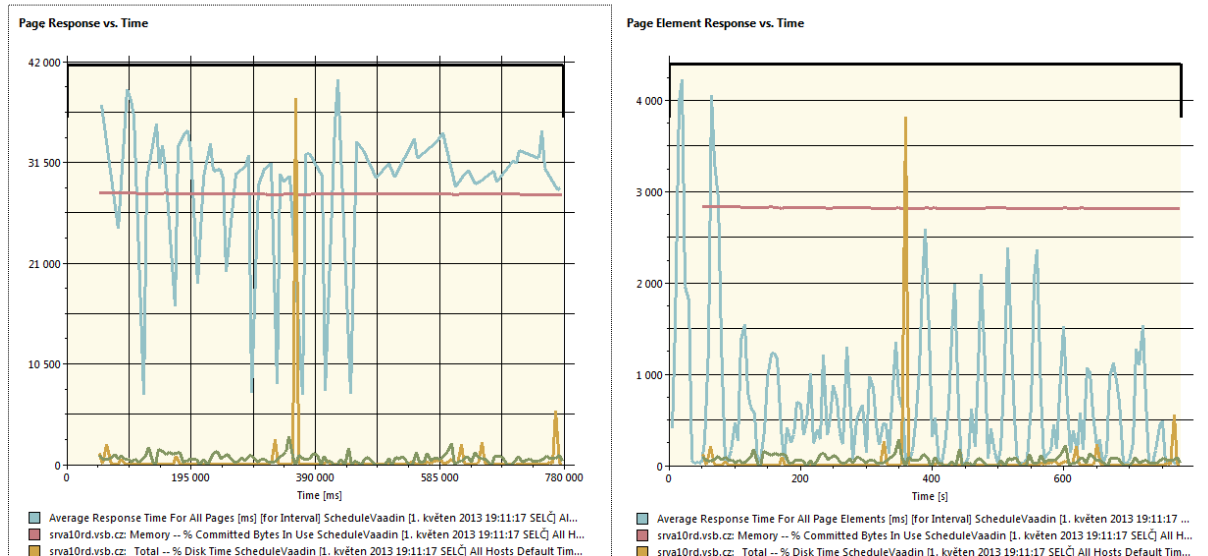
V mé verzi performance testeru jsem byl omezen na 5 uživatelů. Rozdělil jsem uživatele na dvě skupiny. První, která web prohlíží, ta tvoří 75% a 20 iterací. Druhá, která v e-shopu nakupuje a ta tvoří 25% a 5 iterací.

## 11.6.2 Vaadin



Obrázek 20 - Využití CPU, Disku a RAM - Vaadin

# Response vs. Time Summary



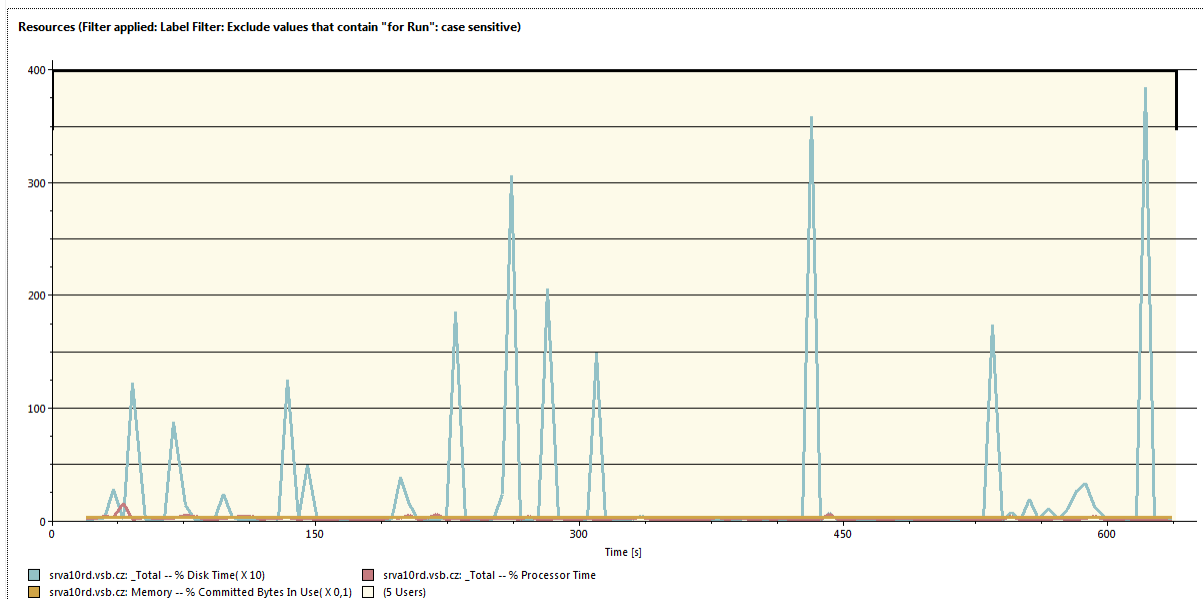
Obrázek 21 - Doba odezvy - Vaadin

Doba odezvy	[ms]
Nejvyšší doba odezvy	41696
Nejnižší doba odezvy	7065
Průměrná doba odezvy	30 120,2
Využití disku	[%]
Nejvyšší využití disku	38,2
Nejnižší využití disku	0
Průměrné využití disku	0,46
Využití procesoru	[%]
Nejvyšší využití CPU	2,89
Nejnižší využití CPU	0
Průměrné využití CPU	0,61
Využití paměti	[%]
Nejvyšší využití RAM	28,2
Nejnižší využití RAM	28,1
Průměrné využití RAM	28,4

Tabulka 5 - Výkonnostní a zátěžové testy Vaadin

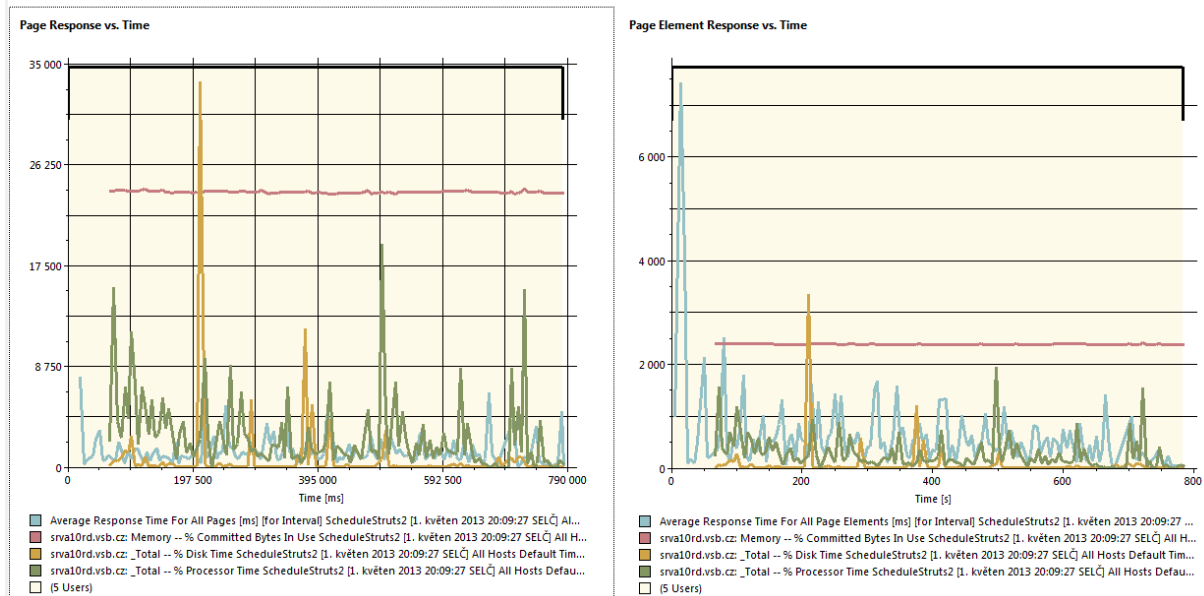
## 11.6.3 Java Server Faces

### Resources



Obrázek 22 - Využití CPU, Disku a RAM - JSF

### Response vs. Time Summary

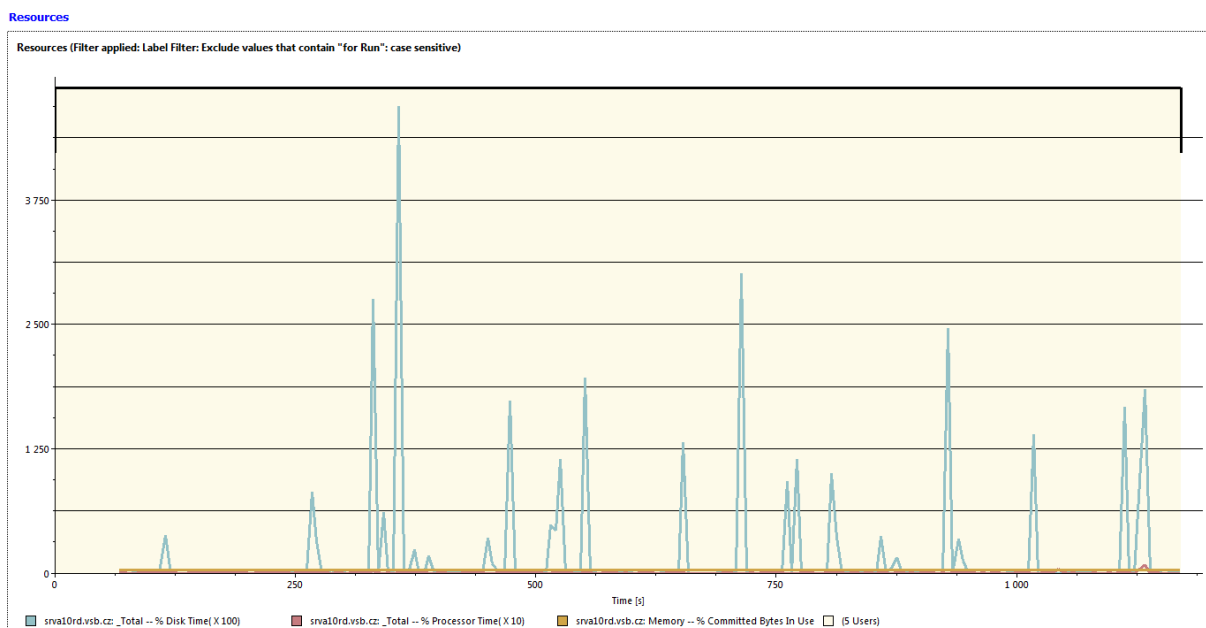


Obrázek 23 - Doba odezvy - JSF

<b>Doba odezvy</b>	<b>[ms]</b>
Nejvyšší doba odezvy	51583
Nejnižší doba odezvy	45
Průměrná doba odezvy	1275,7
<b>Využití disku</b>	<b>[%]</b>
Nejvyšší využití disku	33,4
Nejnižší využití disku	0,01
Průměrné využití disku	0,68
<b>Využití procesoru</b>	<b>[%]</b>
Nejvyšší využití CPU	2,67
Nejnižší využití CPU	0
Průměrné využití CPU	2,61
<b>Využití paměti</b>	<b>[%]</b>
Nejvyšší využití RAM	24,2
Nejnižší využití RAM	23,7
Průměrné využití RAM	23,9

Tabulka 6 - Výkonnostní a zátěžové testy JSF

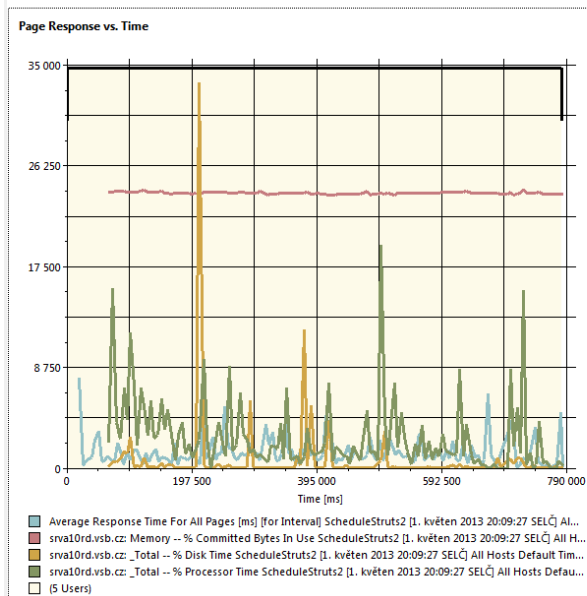
## 11.6.4 Spring MVC



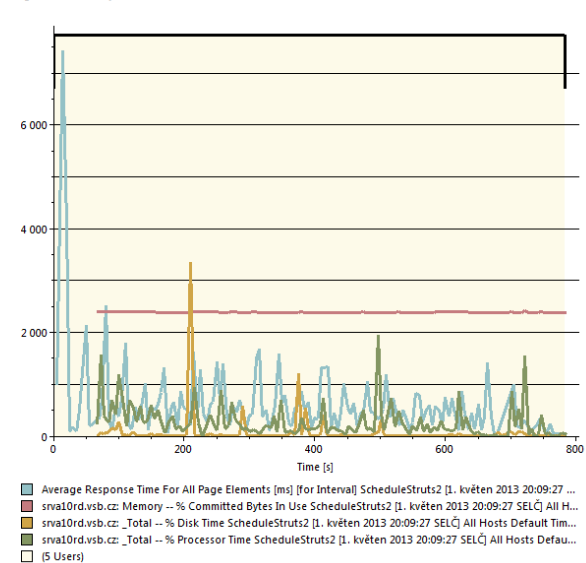
Obrázek 24 - Využití CPU, Disku a RAM – Spring MVC



# Response vs. Time Summary



# Page Element Response vs. Time



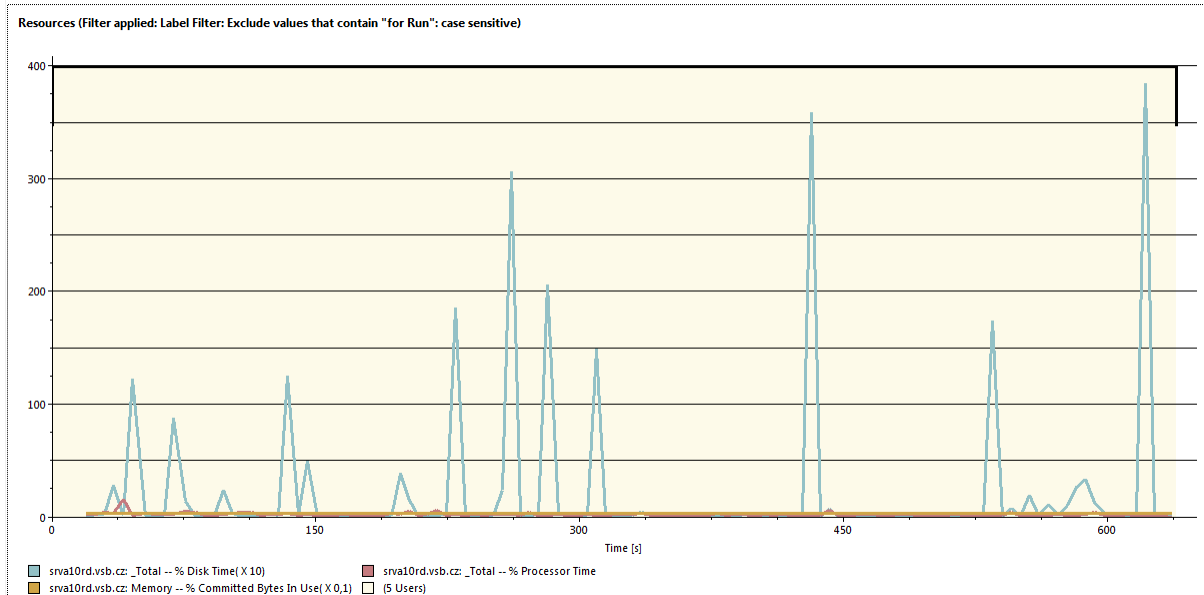
Obrázek 25 - Doba odezvy – Spring MVC

Doba odezvy	[ms]
Nejvyšší doba odezvy	39651
Nejnižší doba odezvy	59
Průměrná doba odezvy	2181,4
Využití disku	[%]
Nejvyšší využití disku	46,9
Nejnižší využití disku	0
Průměrné využití disku	1,61
Využití procesoru	[%]
Nejvyšší využití CPU	8,22
Nejnižší využití CPU	0
Průměrné využití CPU	0,41
Využití paměti	[%]
Nejvyšší využití RAM	24,2
Nejnižší využití RAM	27
Průměrné využití RAM	27,1

Tabulka 7 - Výkonnostní a zátěžové testy Spring MVC

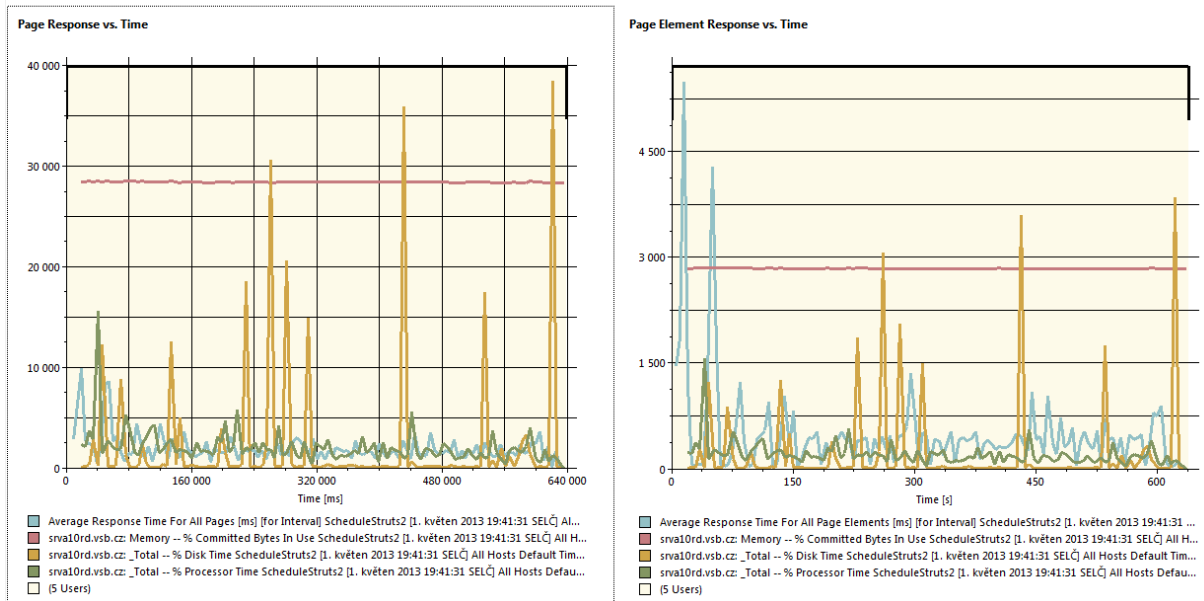
## 11.6.5 Struts2

### Resources



Obrázek 26- Využití CPU, Disku a RAM – Struts2

### Response vs. Time Summary



Obrázek 27 - Doba odezvy – Struts2

<b>Doba odezvy</b>	<b>[ms]</b>
Nejvyšší doba odezvy	14228
Nejnižší doba odezvy	76
Průměrná doba odezvy	1747
<b>Využití disku</b>	<b>[%]</b>
Nejvyšší využití disku	38,4
Nejnižší využití disku	0,02
Průměrné využití disku	2,18
<b>Využití procesoru</b>	<b>[%]</b>
Nejvyšší využití CPU	15,6
Nejnižší využití CPU	0
Průměrné využití CPU	2,01
<b>Využití paměti</b>	<b>[%]</b>
Nejvyšší využití RAM	28,6
Nejnižší využití RAM	28,4
Průměrné využití RAM	28,4

Tabulka 8 - Výkonnostní a zátěžové testy – Struts2

## 11.6.6 Vyhodnocení

Činnost/Framework	Vaadin	JSF	Spring MVC	Struts2
<b>Doba odezvy</b>	[ms]	[ms]	[ms]	[ms]
Nejvyšší doba odezvy	41696	51583	39651	14228
Nejnižší doba odezvy	7065	45	59	76
Průměrná doba odezvy	30120,2	1275,7	2181,4	1747
<b>Využití disku</b>	[%]	[%]	[%]	[%]
Nejvyšší využití disku	38,2	33,4	46,9	38,4
Nejnižší využití disku	0	0,01	0	0,02
Průměrné využití disku	0,46	0,68	1,61	2,18
<b>Využití procesoru</b>	[%]	[%]	[%]	[%]
Nejvyšší využití CPU	2,89	2,67	8,22	15,6
Nejnižší využití CPU	0	0	0	0
Průměrné využití CPU	0,61	2,61	0,41	2,01
<b>Využití paměti</b>	[%]	[%]	[%]	[%]
Nejvyšší využití RAM	28,2	24,2	24,2	28,6
Nejnižší využití RAM	28,1	23,7	27	28,4
Průměrné využití RAM	28,4	23,9	27,1	28,4
<b>CELKEM</b>	<b>6</b>	<b>9</b>	<b>7</b>	<b>5</b>

Tabulka 9 - Zátěžové a výkonostní testy

## 11.7 Podpora technologie AJAX a uživatelská "přítulnost"

AJAX (Asynchronous Javascript and XML) je technologie, pomocí které měníme obsah webové stránky bez nutnosti jejího opětovného načtení. Tento způsob obnovy dat na stránce je rychlejší, šetrnější k serverovému výkonu a uživatelsky příjemnější než způsob, při kterém znovu načítáme celou stránku. Podíváme se tedy na podporu v jednotlivých frameworkích.

### 11.7.1 Vaadin

Velké množství programátorů používá Vaadin právě pro jeho výbornou podporu AJAXu, díky které můžete využívat celé řady komponent. S Vaadinem se nemusíte starat o AJAX komunikaci mezi prohlížečem a serverem a také o to, jak bude aplikace vypadat v různých verzích prohlížeče

### 11.7.2 Java Server Faces

Co se týká JSF vyzkoušel jsem pár příkladů na práci s AJAXem. Není problém AJAX v tomto frameworku využívat pomocí tagů `f:ajax`.

### 11.7.3 Spring MVC

Ve Springu 3.0. je však nově implementována zjednodušená podpora pro Ajax v podobě `mvc-ajax` modulu.

### 11.7.4 Struts2

Oproti svému předchůdci plně podporuje technologii AJAX. Technologie AJAX je ve struts2 podporována díky DOJO (JavaScriptová knihovna).

### 11.7.5 Vyhodnocení

V této sekci je vyhodnocení velmi obtížné, existuje řada pluginů, které rozšiřují možnosti používání technologie AJAX. Řekl bych, že v tomto jsou si všechny mnou testované frameworky rovny. Mým favoritem je framework Vaadin, který AJAX využívá ve velkém a jeho podpora je zde asi nejlepší.

Činnost/Framework	Vaadin	JSF	Spring MVC	Struts2
Seznámení s technologií	9	7	8	8
<b>CELKEM</b>	9	7	8	8

Tabulka 10 - Podpora technologie AJAX

## 11.8 Počet řádků zdrojového kódu a podpůrných souborů

V této části mého srovnání se zaměřím na počet řádků zdrojového kódu, tedy Javovských souborů, plus velikost view stránek, tedy stránek `.jsp`, `.html`, `.atd.` a v neposlední řadě počet řádků

konfiguračních a podpůrných souborů, nejčastěji tedy XML souborů. Zde by mělo být jednoduché měřítko, čím méně kódu programátor musí napsat tím lépe.

### 11.8.1 Vaadin

Typ souboru	Počet souborů	Počet tříd	Počet metod	Počet řádků
Zdrojový kód (Java)	38	37	288	4001
View kód (jsp, xhtml, html, atd.)	0	X	X	0
Podpůrné soubory (konfigurace, XML, atd.)	3	X	X	39
Ostatní	1	X	X	281
<b>CELKEM</b>	<b>42</b>	<b>37</b>	<b>288</b>	<b>4321</b>

Tabulka 11 - Rozsáhlost kódu Vaadin

## 11.8.2 Java Server Faces

Typ souboru	Počet souborů	Počet tříd	Počet metod	Počet řádků
Zdrojový kód (Java)	21	21	213	3100
View kód (jsp, xhtml, html, atd.)	8	X	X	1186
Podpůrné soubory (konfigurace, XML, atd.)	2	X	X	41
Ostatní	1	X	X	281
<b>CELKEM</b>	<b>32</b>	<b>21</b>	<b>213</b>	<b>4608</b>

Tabulka 12 - Rozsáhlost kódu JSF

## 11.8.3 Spring MVC

Typ souboru	Počet souborů	Počet tříd	Počet metod	Počet řádků
Zdrojový kód (Java)	17	17	174	1643
View kód (jsp, xhtml, html, atd.)	<b>9</b>	X	X	1267
Podpůrné soubory (konfigurace, XML, atd.)	3	X	X	86
Ostatní	1	X	X	281
<b>CELKEM</b>	<b>27</b>	<b>17</b>	<b>174</b>	<b>3277</b>

Tabulka 13 - Rozsáhlost kódu Spring MVC

## 11.8.4 Struts 2

Typ souboru	Počet souborů	Počet tříd	Počet metod	Počet řádků
Zdrojový kód (Java)	19	19	212	2789
View kód (jsp, xhtml, html, atd.)	<b>11</b>	X	X	1876
Podpůrné soubory (konfigurace, XML, atd.)	3	X	X	77
Ostatní	1	X	X	281
<b>CELKEM</b>	<b>34</b>	<b>19</b>	<b>212</b>	<b>5023</b>

Tabulka 14 - Rozsáhlost kódu Struts2

## 11.8.5 Vyhodnocení

Činnost/Framework	Vaadin	JSF	Spring MVC	Struts2
Rozsáhlost kódu	<b>6</b>	<b>7</b>	<b>8</b>	<b>5</b>
<b>CELKEM</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>5</b>

Tabulka 15 - Rozsáhlost kódu

## 11.9 Srovnání kódu – přihlášení do aplikace

V této kapitole se podíváme a zkusíme srovnat složitost přihlášení do aplikace (e-shopu). Ukážeme si jednotlivé kusy kódu a pokusíme se je porovnat.

### 11.9.1 Vaadin

Vaadin, jako jediný z mnou porovnávaných frameworku, neobsahuje žádný formulář na view stránce, ale pouze listener, který samotné logování do aplikace přepoše do této mé třídy, kde už se vykonává samotné rozhodování, zda uživatele do aplikace pustit, či nikoliv.

```
public class LoginFunkce {
    private String login;
    private String heslo;
```



```

private HlavniOkno mainWindow;
public LoginFunkce(HlavniOkno mainWindow, String login, String heslo) {
    this.mainWindow = mainWindow;
    this.login = login;
    this.heslo = heslo;
    init();
}
private void init() {
    boolean log = false;
    UserProfile user = (UserProfile)
    DaoFactory.getInstance().getUser().getById(login);
    if (user != null) {
        if (user.getPassword().equals(heslo)) {
            log = true;
        } else {
            log = false;
        }
    } else {
        log = false;
    }
    //Neprenosna cast
    if (log) {
        mainWindow.setPrihlasenyUzivatel(user);
        mainWindow.showNotification("Přihlášení proběhlo úspěšně");
        List<Object> transakce =
        DaoFactory.getInstance().getTransaction().getAll();
        for (Object object : transakce) {
            Orderss objednavka = (Orderss) object;
            if (objednavka.getStatus().equals("Objednáno")) {
                DaoFactory.getInstance().getTransaction().delete(objednavka);
            }
        }
        mainWindow.init();
    } else {
        mainWindow.showNotification("Nesprávné heslo nebo login!",
        Notification.TYPE_ERROR_MESSAGE);
    }
}
}

```

## 11.9.2 Java Server Faces

JSF obsahuje klasickou view část s formulářem, který se POSTem přesměrovává do java Beany, kde je opět ukrytá logika aplikace.

### 11.9.2.1 View část – xhtml

```

<div id="login">
    <h:form id="yes" rendered="#{login.logged}">
        #{login.jmeno} <h:commandLink value="Odhlásit"
        action="#{login.logout()}" /><br /><br />
    </h:form>
</div>

```

```

        <a href="kosik.xhtml?param=#{login.jmeno}">Košík</a><br /><br />
</h:form>

<h:form id="no" rendered="#{!login.loged}">
    <b>Jméno</b> <br />
    <h:inputText required="true" value=#{login.jmeno}" title="login"
id="jmeno" /> <br />
    <b>Heslo</b> <br />
    <h:inputSecret required="true" value=#{login.heslo}"
title="login" id="pasw" /> <br /> <br />
    <h:commandButton value="Přihlásit" action=#{login.login()}" />
</h:form>
</div>

```

### 11.9.2.2 Java Beana

```

public class Login {
    private boolean loged;
    private String jmeno;
    private String heslo;
    private FacesContext context = (FacesContext)
FacesContext.getCurrentInstance();
    private HttpSession session = (HttpSession)
context.getExternalContext().getSession(false);
    public Login() {
        loged = false;
    }
    public String logout() {
        loged = false;
        if (session != null) {
            session.setMaxInactiveInterval(3);
        }
        return "index.xhtml";
    }
    public String login() {
        try {
            if (jmeno != null && heslo != null) {
                UserProfile user = (UserProfile)
DaoFactory.getInstance().getUser().getById(jmeno);
                if (user.getPassword().equals(heslo)) {
                    loged = true;
                } else {
                    loged = false;
                }
            }
        } catch (Exception e) {
            loged = false;
        }
        return "index.xhtml";
    }
}

```

### 11.9.3 Spring MVC

Opět klasická struktura tedy view stránka s formulářem pro zadání přihlašovacích údajů. Java třída pro logiku přihlašování, v našem případě se jedná o AbstractFormController.

#### 11.9.3.1 View část

```
<h2>Přihlášení</h2>
    <form:form commandName="user" method="post"
        action="login.htm">

        Login: <input type="text" name="login"><br/>
        Heslo: <input type="text" name="heslo"><br/>
        <input type="submit" value="Submit">
    </form:form>
```

#### 11.9.3.2 Java class

```
public class Login extends AbstractFormController {

    @Override
    protected ModelAndView showForm(HttpServletRequest request hsr,
        HttpServletResponse hsr1, org.springframework.validation.BindException be)
        throws Exception {
        return null;
    }

    @Override
    protected ModelAndView processFormSubmission(HttpServletRequest request hsr,
        HttpServletResponse hsr1, Object o,
        org.springframework.validation.BindException be) throws Exception {
        User user = (User) o;
        boolean logged=false;
        UserProfile userDb = (UserProfile)
        DaoFactory.getInstance().getUser().getById(user.getLogin());
        if (userDb.getPassword().equals(user.getHeslo())) {
            logged = true;
        } else {
            logged = false;
        }
        ModelAndView model;
        if (logged) {
            model = new ModelAndView("login");
            model.addObject("login", user.getLogin());
            model.addObject("heslo", user.getHeslo());
        } else {
            model = new ModelAndView("error");
            model.addObject("login", user.getLogin());
            model.addObject("heslo", user.getHeslo());
        }
        return model;
    }
}
```

#### 11.9.3.3

## 11.9.4 Struts2

Osobně bych řekl, že Struts2 je svou strukturou velmi podobné Java Server Faces, liší se pouze v dodatečné konfiguraci XML souborů.

### 11.9.4.1 View část – JSP

```
<s:if test="%{#session.logged!= 'true'}">
    <h2>Přihlášení</h2>
    <s:form action="login.action" method="post">
        <s:textfield name="username" size="20" />
        <s:password name="password" size="20" />
        <s:submit method="execute" value="Přihlášení" align="left" />
    </s:form>
</s:if>
<s:elseif test="%{#session.logged== 'true'}">
    <h2>Přihlášen</h2><br>
    <b><s:property value="#session.name" /></b><br><br>
    <b><s:property value="#session.email" /></b><br><br>
    <s:form id="kosikButton" action="kosik.action">
        <s:submit value="Košík" type="button"/>
    </s:form>
    <s:form id="odhlasitButton" action="logout.action">
        <s:submit value="Odhlásit" type="button"/>
    </s:form>
</s:elseif>
```

### 11.9.4.2 Java class

```
public class LoginAction extends ActionSupport {
    private String username;
    private String password;
    private String online;
    public String execute() throws Exception {
        UserProfile user;
        if (username != null && password != null) {
            user = (UserProfile)
                DaoFactory.getInstance().getUser().getById(username);
            try {
                user.getPassword();
            } catch (Exception e) {
                return ERROR;
            }
            if (user.getPassword().equals(password)) {
                Map session = ActionContext.getContext().getSession();
                session.put("logged", "true");
                session.put("id", user.getUserId());
                session.put("name", user.getName());
                session.put("email", user.getEmailId());
                List<Object> transakce =
                    DaoFactory.getInstance().getTransaction().getAll();
                for (Object object : transakce) {
```

```

        Orderss objednavka = (Orderss) object;
        if (objednavka.getStatus().equals("Objednáno")) {
            DaoFactory.getInstance().getTransaction().delete(objednavka);
        }
    }
    return SUCCESS;
} else {
    return ERROR;
}
}
return ERROR;
}

```

### 11.9.5 Vyhodnocení

Dle mého názoru je nejelegantnější řešení to, které poskytuje Vaadin bez nutnosti vytvoření view stránky, proto získal v našem hodnocení nejvíce bodů. Druhý se umístil framework JSF a hned za ním framework Struts2, a to jen proto, že zde přibyla konfigurace XML souboru pro každou actions (login, logout). Nejvíce jsem se natrápil s logováním do aplikace ve Springu, kde jsem dlouho bojoval se sessions, proto se tento framework umístil v mém hodnocení na posledním místě.

Činnost/Framework	Vaadin	JSF	Spring MVC	Struts2
Seznámení s technologií	8	7	5	6
<b>CELKEM</b>	<b>8</b>	<b>7</b>	<b>5</b>	<b>6</b>

## 12 Celkové vyhodnocení porovnávaných frameworků

Nyní se podíváme na celkové hodnocení všech testovaných frameworků formou tabulky, tak aby bylo možné rychle zjistit, jak si daný framework vedl. Tato tabulka by měla programátorovi usnadnit výběr frameworku pro danou aplikaci. Rovněž provedu slovní hodnocení každého z frameworků.

### 12.1 Slovní hodnocení

Závěrečné zhodnocení všech testovaných frameworků bude vztaženo k naší testovací aplikaci, kterou byl již výše zmíněný e-shop.

Jak vidíme z níže přiložené tabulky, celkovým vítězem našeho srovnání se stal framework Java Server Faces. Naopak na nejnižší příčce a tedy poslední se umístil framework Struts2. Nelze ale jednostranně říci, že i když framework xxx „vyhrál“ naše srovnání, je vhodný na všechny typy aplikací. Pokud se podíváme podrobně na tabulku 4, zjistíme, že každý framework vyniká v něčem jiném a takto na něj musí být pohlíženo při jeho výběru.

#### 12.1.1 Vaadin

Vaadin je ze všech čtyř frameworků, které mi prošly klávesnicí, nejspecifičtější. Je odlišný od běžných frameworků a takto se na něj musí pohlížet. Obrovskou výhodou tohoto frameworku, jsou jeho komponenty a dostupné funkce, které jsou ze všech frameworků na nejvyšší úrovni. Bohužel nevýhodou toho frameworku je rychlost, ve které zaostává celkem zásadně za všemi ostatními frameworky.

#### 12.1.2 Java Server Facer

Osobně jsou pro mě výsledky toho frameworku velkým překvapením, v minulosti jsem se s ním již setkal, ale velký dojem na mě neudělal. Tento názor jsem změnil, s Java Server Face se mi tentokrát pracovalo velmi dobře, rovněž jeho výsledky v testech jsou dobré, ne-li výborné. Lze asi tvrdit, že tento framework se stal pomyslným vítězem mého srovnání.

#### 12.1.3 Spring MVC

Framework od kterého jsem toho hodně očekával, a to pro jeho dobrou pověst a doporučení mých ostatních programátorských kolegů. Bohužel musím říci, že pro mě je programování v tomto frameworku zklamáním a očekával jsem od něj více. Je možné, že je to jen můj subjektivní pocit, ale ani výsledky mých srovnání nijak zásadně tento framework nevyzvedávají. Lze snad jen zmínit dobré výsledky, co se týká rychlosti při velkém počtu uživatelů zatěžujících stránku.

## 12.1.4 Struts2

Další z frameworků, který byl pro mě novinkou. Dle mého názoru je možno říci, že frameworky Spring MVC a Struts2 jsou si velmi podobné, což asi nikoho nepřekvapí. Co ale překvapuje mě, je fakt, že v tomto frameworku se mi vyvíjelo lépe, než v frameworku Spring MVC, který má právě ze Struts2 vycházet a odstraňovat jeho nedostatky. Je možné, že při vývoji jiné (např. složitější, větší) aplikace by tyto rozdíly byly citelnější a výhody frameworku Spring MVC by vyšly najevo. Toto se ale v mém případě neudálo. Nevýhodu frameworku Struts2 vidím například ve složitém systému XML souborů, na který si ale lze rychle zvyknout.

## 12.2 Tabulkové vyhodnocení

Činnost/Framework	Vaadin	JSF	Spring MVC	Struts2
Složitost vývoje	7	8	6	6
Náročnost prvotního seznámení	5	8	7	6
Dokumentace	9	7	8	4
Podpora ve vývojových nástrojích	9	8	8	8
Zátěžové a výkonnostní testy	6	9	7	5
Podpora AJAX	9	7	8	8
Rozsáhlost kódu	6	7	8	5
Ukázka kódu (Přihlašování do aplikace)	8	7	5	6
CELKEM	59	61	57	48

Tabulka 16 – Celkové vyhodnocení

## 13 Závěr

Na závěr bych chtěl shrnout veškeré porovnávání a vyhodnocování daných frameworků, ale rovněž vývoj testovací aplikace, dojmy z analýz a ostatní věci.

Cílem mé práce bylo poskytnout tápajícímu vývojáři jednoduchý rozcestník, podle kterého by si mohl vybrat a nejlépe odhadnout, který webový framework by byl pro jeho práci nejvhodnější. Při vývoji aplikace jsem se nesetkal se velkým problémem, který by mi nějak zásadně znemožnil vývoj testovací aplikace. Z toho vyplývá, že se podařilo e-shop úspěšně implementovat ve všech mnou zvolených frameworkcích.

Jak již bylo řečeno, nelze paušálně tvrdit, že by jeden framework byl vyloženě špatný a druhý ve všem vynikal. Je třeba pečlivě zvážit kritéria výběru podle vlastností zamýšlené webové aplikace a k tomuto by měla sloužit tato práce. Doufám, že v ní vynikly přednosti a zároveň nedostatky jednotlivých frameworků. Rovněž věřím, že přínosem budou výkonnostní a zátěžové testy, které byly spuštěny nad všemi testovacími aplikacemi a které dokážou poskytnout hmatatelné a konkrétní výsledky pro každý framework.

Práce pro mě byla přínosem v tom směru, že jsem se konečně setkal s frameworky, s kterými jsem doposud neměl žádné zkušenosti, jmenovitě tedy s Spring MVC a Struts2. Naučil a pochopil jsem nové technologie a postupy, které tyto frameworky využívají a doufám, že toto v budoucnu uplatním v praxi.



## 14 Seznam tabulek

Tabulka 1 - Složitost a rychlost vývoje .....	28
Tabulka 2 - Náročnost prvotního seznámení s technologií .....	30
Tabulka 3 - Dostupnost a přehlednost dokumentace.....	31
Tabulka 4 - Podpora v nástrojích pro vývoj .....	32
Tabulka 6 - Výkonnostní a zátěžové testy Vaadin .....	34
Tabulka 5 - Výkonnostní a zátěžové testy JSF.....	36
Tabulka 7 - Výkonnostní a zátěžové testy Spring MVC .....	37
Tabulka 8 - Výkonnostní a zátěžové testy – Struts2 .....	39
Tabulka 9 - Zátěžové a výkonnostní testy .....	40
Tabulka 10 - Podpora technologie AJAX .....	41
Tabulka 11 - Rozsáhlost kódu Vaadin.....	42
Tabulka 12 - Rozsáhlost kódu JSF .....	43
Tabulka 13 - Rozsáhlost kódu Spring MVC .....	43
Tabulka 14 - Rozsáhlost kódu Struts2.....	44
Tabulka 15 - Rozsáhlost kódu .....	44
Tabulka 16 – Celkové vyhodnocení.....	51

## 15 Seznam obrázků

Obrázek 1, MVC .....	3
Obrázek 2 - Spring struktúra .....	5
Obrázek 3 - Vaadin architektura .....	7
Obrázek 4 - Struts2 architektura.....	10
Obrázek 5 - JSF architektura .....	13
Obrázek 6 - Schéma databáze .....	16
Obrázek 7 - Úvodní strana, Vaadin .....	19
Obrázek 8, Nákupní košík, Vaadin .....	20
Obrázek 9 - Registrace, Vaadin.....	20
Obrázek 10 - Úvodní strana, JSF.....	21
Obrázek 11 - Registrace, JSF .....	22
Obrázek 12 - Košík, JSF.....	22
Obrázek 13 - Úvodní strana, Struts2 .....	23
Obrázek 14 - Registrace, Struts2.....	24
Obrázek 15 - Košík, Struts2 .....	24
Obrázek 16 - Úvodní strana, Spring.....	25
Obrázek 17 - Nákupní košík, Spring .....	26
Obrázek 18 - Registrace, Spring.....	26
Obrázek 19 - Virtuální server .....	32
Obrázek 20 - Využití CPU, Disku a RAM - Vaadin .....	33
Obrázek 21 - Doba odezvy - Vaadin .....	34
Obrázek 22 - Využití CPU, Disku a RAM - JSF.....	35
Obrázek 23 - Doba odezvy - JSF.....	35
Obrázek 22 - Využití CPU, Disku a RAM – Spring MVC .....	36
Obrázek 23 - Doba odezvy – Spring MVC .....	37
Obrázek 24- Využití CPU, Disku a RAM – Struts2.....	38
Obrázek 25 - Doba odezvy – Struts2.....	38

## Literatura

1. **Spring.** Spring source. *Spring source*. [Online] [Cited: duben 2, 2013.] <http://www.springsource.org/>.
2. **Oracle.** JavaServer Faces Technology. *Oracle*. [Online] 2013. <http://www.oracle.com/technetwork/java/javaee/jaserverfaces-139869.html>.
3. **Apache.** The Apache Software Foundation. *About Apache Struts 2*. [Online] [Cited: duben 2, 2013.] <http://struts.apache.org/development/2.x/>.
4. **Vaadin.** thinking of U and I. *Vaadin*. [Online] [Cited: duben 12, 2013.] <https://vaadin.com/home>.
5. **Oracle.** Java EE at a Glance. *Oracle*. [Online] [Cited: duben 2, 2013.] <http://www.oracle.com/technetwork/java/javaee/overview/index.html>.
6. **Wikipedia.** Model-View-Controller. *Wikipedie*. [Online] [Cited: duben 1, 2013.] <http://cs.wikipedia.org/wiki/Model-view-controller>.
7. **Johnson, Rod.** *Expert One-on-One J2EE Design and Development*. s.l. : Wrox, 2002.
8. **Spring.** Spring source community. *Documentation*. [Online] [Cited: duben 11, 2013.] <http://www.springsource.org/documentation>.
9. **Google.** Google Web Toolkit. *Google Developers*. [Online] [Cited: duben 11, 2013.] <https://developers.google.com/web-toolkit/?hl=cs>.
10. **Grönroos, Marko.** *Book of Vaadin*. 2005.
11. **Apache.** Apache Struts 2 Documentation. *Apache*. [Online] [Cited: březn 22, 2013.] <http://struts.apache.org/release/2.0.x/docs/home.html>.
12. **Wikipedia.** JavaServer Pages. *Wikipedia*. [Online] [Cited: březn 23, 2013.] [http://en.wikipedia.org/wiki/JavaServer\\_Pages](http://en.wikipedia.org/wiki/JavaServer_Pages).